DOCUMENT RESUME

ED 301 458 SE 050 178

AUTHOR	Miller, Samuel K.
TITLE	Selecting and Implementing Educational Software.
REPORT NO	ISBN-0-205-10468-1
PUB DATE	87
NOTE	189p.; Drawings and some small print may not reproduce well.
AVAILABLE FROM	Allyn & Bacon/Logwood Division, 160 Gould Street, Needham Heights, MA 02194-2310 ($26.95, 20% off 10 or more).
PUB TYPE	Books (010) -- Reports - Descriptive (141) -- Book/Product Reviews (072)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; Computer Literacy; *Computer Uses in Education; *Courseware; Elementary Secondary Education; Higher Education; Inservice Teacher Education; Instructional Materials; Microcomputers; *Preservice Teacher Education; Resource Materials; Science Education; *Science Instruction; Teacher Improvement
IDENTIFIERS	Computer Resources

ABSTRACT

This book is a nontechnical introduction to the use and understanding of computers and educational software. Topics included in the book are: (1) uses of computers in today's educational system; (2) a basic description of computers and hardware; (3) a basic understanding of computer-assisted instruction; (4) programming aids and tools; (5) how to locate useful educational software; (6) evaluating software; and (7) using and organizing educational software. Each chapter includes discussion questions and resource lists. (MVL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED301458

SE 050 178

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
David M.
Haar
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Selecting and Implementing Educational Software

Samuel K. Miller

ALLYN AND BACON, INC.
Boston London Sydney Toronto

Copyright © 1987 by Allyn and Bacon, Inc.
7 Wells Avenue, Newton, Massachusetts 02159

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

Library of Congress Cataloging-in-Publication Data

Miller, Samuel K.
Selecting and implementing educational software.

Bibliography: p.
Includes index.
1. Education—United States—Curricula—Computer programs. 2. Education—United States—Curricula—Computer programs—Evaluation. 3. Computer-assisted instruction—United States. I. Title.
LB1028.68.M55 1987 370'.2854'25 86-28841
ISBN 0-205-10468-1

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1 92 91 90 89 88 87

Contents

Preface *vii*

1 Instructional Software: Issues and Concerns 1

Introduction 1

Comparing Printed Materials and Microcomputer Software 2

Microcomputers and Teacher Education 11

Other Factors Related to Educational Software 13

Conclusion 15

Readings and Resources 16

References 16

Projects and Questions 17

2 Overview of CAI Software 19

Introduction 19

A Brief History of Computers 19

Computer Basics 20

Computer Assisted Instruction 25

Drill and Practice 26

Tutorial CAI 32

Simulation-Oriented CAI 42
Instructional Games 49
Conclusion 55
Readings and Resources 56
References 58
Projects and Questions 59

3 **Tool Software** 61
Introduction 61
Classifying Tool Programs 62
General-Purpose Tools 62
Special-Purpose Tools 74
Tool-Making Tools 83
Conclusion 90
Readings and Resources 90
References 92
Projects and Questions 92

4 **Locating Educational Software** 95
Introduction 95
Beginning the Search for Software 95
Obtaining Software to Preview 106
Conclusion 108
Readings and Resources 108
Projects and Questions 118

5 Conducting Software Evaluations and Reviews 121

Introduction 121

Selection Criteria and Evaluation Forms 121

The Evaluation and Review Process 122

Evaluation Questions 132

Conclusion 153

Readings and Resources 155

References 158

Projects and Questions 158

6 Organizing and Using Educational Software 161

Introduction 161

Organizing a Software Collection 161

Organizing the Learning Environment 168

Conclusion 177

Readings and Resources 178

References 178

Projects and Questions 178

Index 181

Preface

Computer instruction has become an integral part of American education during the past several years. Successful use of the computer in education is dependent upon the value of the available software. To determine which software programs merit use in the classroom, to make wise investments with limited funds, and to learn how to organize and manage the computer as an adjunct to instruction are new skills for most teachers. I wrote *Selecting and Implementing Educational Software* with the belief that all teachers must develop a philosophy for selecting and using computer software predicated on instructional concerns.

This book presents a lot of practical information about computers in a nontechnical way. The scope of the book excludes technical details about the operation of computers, hardware selection, and materials that are simply too difficult or too expensive for most students and their teachers to use.

Chapter 1 focuses on issues related to factors that influence software selection—how teachers select and use print and computer materials. The factors that influence the quality and price of software also are addressed in the first chapter.

Chapters 2 and 3 present information and examples of popular educational software. An historical background leading to the development of today's educational software is included. Wherever possible, research is presented about the instructional effectiveness of specific categories of computer software. Each chapter emphasizes instructional considerations and features that characterize "good" software.

Chapter 4 considers the acquisition and evaluation of educational software. Various approaches to software evaluation are presented in Chapter 5 along with examples of evaluation forms that are used by experienced software evaluators.

Chapter 6 provides guidelines for organizing and implementing a software collection in a school. Because situations vary from school to school, several software management schemes are suggested.

Each chapter includes discussion questions and resource lists for use with preservice and inservice education classes. Chapters from the text have been used by a number of teachers throughout Oregon and Washington. They have demonstrated that teachers have a strong desire to stay abreast of new developments in education. This book is dedicated to them and their colleagues who attend voluntary classes during the summer, on weekends, and after long, hard days at work. It is also dedicated to my wife Zola Ehlers Miller for her assistance throughout the development of this book. Without her patience, intelligence, and critical support, this book could never have come to be.

1

Instructional Software: Issues and Concerns

INTRODUCTION

Imagine that you're a passenger on a space shuttle fueled with regular gasoline instead of rocket fuel and that the pilot of the shuttle is only qualified to fly a crop duster. You would probably have some legitimate concerns about how this sophisticated vehicle will successfully operate! Computer education in the 1980s has similar concerns. The current generation of small, desktop computers, called microcomputers, provides a new tool that many educators believe may be used to enhance every student's education, but learning how to appropriately "fuel" and "pilot" the microcomputer for instructional use is not a simple task.

During the past several years, dramatic changes have occurred in computer technology. Changes in three factors—size, cost, and the computational power of computers—have made educators eager to purchase microcomputers for instructional purposes. As the number of microcomputers in schools continues to grow, the corresponding demand for quality educational software increases. "Software" refers to a computer program, usually stored on a diskette, tape, or cartridge. The demand for software in the areas of instruction, administration, research, and scientific applications has created a multimillion dollar market for software developers and companies. The large number of software products on the market (with a corresponding range in quality) make the selection of educational software a formidable task. The following sections briefly describe some major problems related to computer education and instructional software.

COMPARING PRINTED MATERIALS AND MICROCOMPUTER SOFTWARE

The Influence of Curriculum Materials

Printed curriculum materials tend to determine the objectives and content of instruction for a teacher. While printed curriculum materials may influence the objectives and content that are taught, teachers do modify the *presentation* of content in instructional materials; however, if students have difficulty understanding what the computer is presenting, the computer program cannot be easily modified. This is an important distinction between printed instructional materials and instructional software. Modifying printed materials is relatively easy compared to modifying software because software consists of coded steps or instructions that direct the computer to perform a task. These instructions, which make up a program, are written in a programming language. For example, the microcomputer program listed on page 3 is a small section of a longer program designed to test a student's knowledge of U.S. states and their capitals (Kamins & Waite, 1982). It is written in BASIC, a popular programming language. Can you interpret and modify these commands?

```
865   VTAB 15: HTAB 1:  CALL  -  958
870   IF HS > BS THEN FOR X = CT% - (1 + LEN (NAME$) + 5)
      TO 1 STEP - 1: HTAB X:  CALL - 868: PRINT NAME$"
      WINS! ";: NEXT X: PRINT  :  GOTO 890
      REM BEL IMBEDDED IN LOOP
875   IF HS = BS THEN TITLE$ =  "IT'S A DRAW!!":  FLASH :
      GOSUB 34000:  NORMAL: GOTO 890:
      REM  10 BELS IMBEDDED HERE
880   HTAB 23: PRINT  "THE BEAST WINS!": FOR X = 1 TO 10:
      FOR RAZZ = 1 TO 10:
      J =  PEEK  ( - 16336):J = PEEK  ( - 16336):
      NEXT RAZZ: NEXT X
890   NORMAL
```

If you're a skilled programmer the answer is probably yes. If you're a typical classroom teacher with no programming expertise, the answer is probably no.

The point is that most educational software designed to teach students has very limited flexibility. In most instances, what is in a microcomputer program is exactly what a student is taught.

Time Spent Interacting with Instructional Materials

Studies indicate that teachers and students structure between 80 and 95 percent of their classroom time and 90 percent of their homework around instructional materials (Woodbury, 1979, p. 16). These statistics clearly indicate the significant role of printed curriculum materials. At the present time, students do not spend nearly as much time interacting with educational software as they do with printed curriculum materials.

A comprehensive survey conducted by Johns Hopkins University Center for Social Organization of Schools (Becker, 1985) summarized the extent to which students in 2,300 schools are currently using microcomputers in the classroom. The results indicate that more than one million computers are being used by fifteen million students and some half million teachers in the United States. The majority of computers appear in secondary schools where they are used primarily for programming and computer literacy instruction, while in elementary schools computer assisted instruction remains the computer's primary function.

The survey results indicate that, on the average, an elementary school student uses a microcomputer less than 30 minutes a week; secondary school users have access to a microcomputer for more than 45 minutes per week. Although the amount of time students currently spend using a microcomputer is limited, increased time can be expected as schools continue to purchase equipment.

Field Testing and Revision of Curriculum Materials

Some educators are surprised to learn that the software they've purchased for classroom use has not been systematically tested with students. So, when a student becomes frustrated or bored by a program, it's a sure sign that the developers have based the program's content on their *assumptions* of how students should react to the program.

Who is responsible for the quality of instruction that a student receives? One viewpoint is that teachers are accountable. Another perspective, Learner Verification and Revision (LVR), places the responsibility with the developers of curriculum materials. LVR involves testing the effectiveness of curriculum materials by using them with a sample of students for whom they are intended and then making necessary changes on the basis of those tests. Since 1971, Educational Products Information Exchange (EPIE) has actively promoted LVR as a method for improving the quality of instructional materials.

Software producers have responded to concerns about field testing in various ways, but the industry has not adopted a consistent interpretation of what field testing is or how it should be conducted. Publishers frequently use the "in-house" method of field testing. In-house field testing, sometimes called alpha-testing, typically consists of observing the responses of individuals while they interact with a microcomputer program. The participants in these sessions might be students or even the program developers themselves. This kind of field testing is usually conducted at the publishing company rather than in a school setting.

Another approach to field testing is beta-testing, which involves soliciting appropriate users to try a program in a real-life testing environment. Beta-testers, provided with free software and sometimes a small monetary stipend, are asked to check a program for ease of use and potential errors. Beta-test reports are usually derived from responses to questionnaires by individuals who use a program.

A true field test of educational software should include the use of a microcomputer program under conditions that approximate those for which the software is designed. Data collection—in the form of observations, interviews, questionnaires, and verification that a program does what it purports to do—is essential for a rigorous field test.

Whether or not software producers can afford to use more rigorous field testing procedures is another matter. Rapid changes in microcomputer technology do not always allow for the development of products that are honed to perfection. Many software producers are genuinely interested in developing quality products, but educators must remember that software producers are also in business to make money.

What Research Tells Us about Field Testing

Who is responsible for the quality of instructional products? How much responsibility must software publishers assume to insure that software is thoroughly field tested and revised? The excerpt below is from *EPIEgram*, a monthly newsletter published by the Educational Products and Information Exchange (1984). What do you think? Should the LVR process be mandated for all instructional software?

> Of all the many hundreds of software problems that pass through EPIE's courseware evaluation process, analysts have found few whose documentation indicates that field testing was part of a program's development. Of the over 450 software programs that have been rigorously evaluated by EPIE, fewer than 10 percent indicate that field testing was carried out with users. And of that 10 percent, perhaps only 1 percent could be said to have undergone systematic revision as a result.

Of course, the documentation EPIE reviews as part of its evaluation process doesn't necessarily reveal the whole story of industry approaches to field testing. It was therefore helpful to find that a recent survey of field testing practices of software producers supported our general findings. Carol Truett, professor of library science at the University of Hawaii, surveyed over 406 software producers during 1983, of which only 14 percent responded. It can be presumed that producers who *do* field testing responded to Truett at a higher rate than those who do not, so it would seem unlikely that the 70 percent figure of respondents who indicated that they incorporate field testing into their development process could be generalized to the industry itself. But Truett's respondents do demonstrate the range of interpretations producers apply to the term "field testing." Of those who reported field testing their products, 40 percent relied on teacher evaluations, 11 percent on pre- and post-tests, 2 percent on student evaluations, and 29 percent on other techniques to measure a program's effectiveness. Most respondents used between 3 to 6 teachers and 21 to 50 students to test their software. Most of the testing took place in schools. Producers also listed additional methods for evaluating their programs, the most popular being in-house review and expert review.

From *EPIEgram*, 1984, pp. 1–7.

Locating Curriculum Materials

Common sources of information about printed curriculum materials include catalogs, conferences, presentations from publishers' representatives, and word-of-mouth suggestions from fellow educators. There are hundreds of catalogs of instructional materials and thousands of instructional products—and the flood of educational products is not restricted to printed materials. Estimates have placed the number of companies selling microcomputer software at about 1,800 (Kolata, 1983).

A major problem in the search for instructional software is one of acquiring materials to review. When a new textbook is published, most companies send free samples of the textbook or teacher's guide to schools and selected teachers; the donation of one free book for a classroom order of 30 books is considered by publishers to be a wise investment. However, if software publishers send their products "on approval" or provide free samples, they run the risk of losing a sale because a disk can be readily copied.

Software Piracy

The illegal reproduction of copyrighted software is a complex problem and is not unique to education. Computer software is usually sold in the form of floppy disks or diskettes, which are flat, circular pieces of plastic the size of a 45-rpm record. A diskette retains coded information that provides operating instructions to the computer (see Figure 1.1). A blank diskette, which costs a few dollars, can be used to copy information from another diskette in a matter of minutes; however, the unauthorized copying of a microcomputer program is illegal.

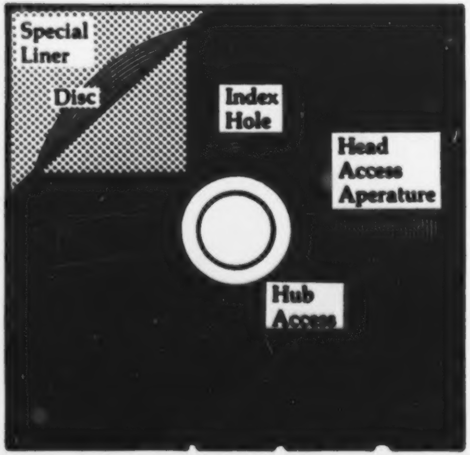
To combat the illegal copying of software (sometimes referred to as pirating), manufacturers have attempted to copy-protect their software by altering the way it is recorded on a diskette, thereby "locking" the program. Copy-protection schemes have in turn created a market for software programs that "unlock" copy protected software. Educators are aware that these programs exist and sometimes use them to copy a master disk and then allow their students to use the copies.

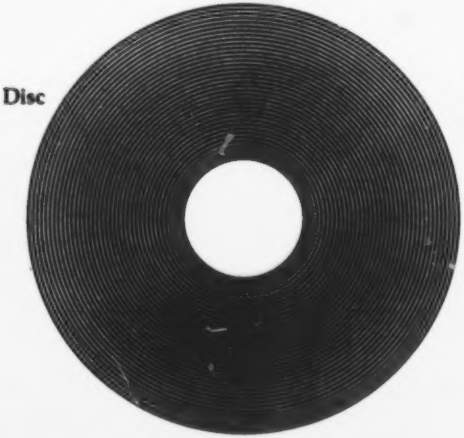
Backup Policies

The term "backup copy" refers to a second copy of a program to be kept in case the original is damaged or lost. Having a backup copy of an educational program is extremely important; educators cannot afford to risk the loss of valuable instructional time or data because their software is lost or destroyed. Software publishers have adopted various policies regarding the availability of duplicate software for backup use. Some publishers provide a backup diskette with a program when it is purchased, while others distribute software that is not copy-protected so that buyers can legally make their own backup copies. Most publishers will sup-

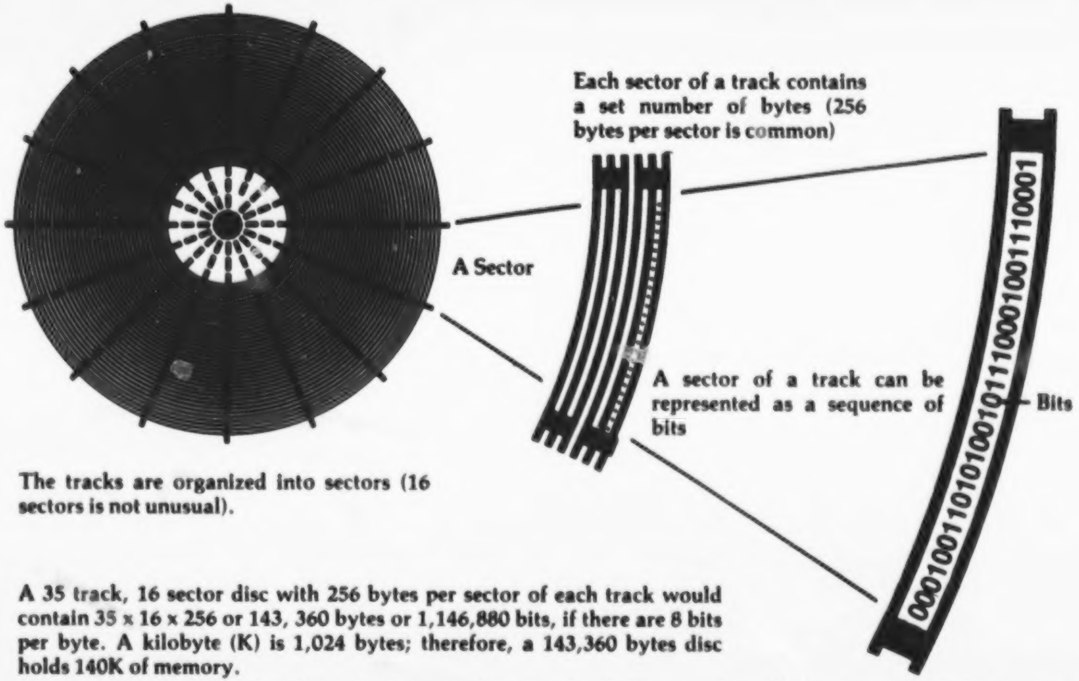
FIGURE 1.1

Information Storage on a Disc


Special Liner
Disc
Index Hole
Head Access Aperature
Hub Access



The information on a disc is stored in tracks. The tracks are concentric circles (35 tracks is common).



Each sector of a track contains a set number of bytes (256 bytes per sector is common)

A sector of a track can be represented as a sequence of bits

The tracks are organized into sectors (16 sectors is not unusual).

A 35 track, 16 sector disc with 256 bytes per sector of each track would contain 35 x 16 x 256 or 143, 360 bytes or 1,146,880 bits, if there are 8 bits per byte. A kilobyte (K) is 1,024 bytes; therefore, a 143,360 bytes disc holds 140K of memory.

Before a disc can be used, it has to be formatted or initialized. This involves placing zeroes in all the memory positions (i.e., in the tracks and sectors). Because different computers vary, the number and placement of their tracks and sectors that are initialized for one computer will not run another with a different track and sector organization.

From *Microcomputer Applications in the Classroom* by Alan Hofmeister, Copyright © 1984 by CBS College Publishing.
Reprinted by permission of CBS College Publishing.

ply a new copy of a diskette if it is ruined, but they usually require the user to return the original diskette—and they charge a replacement fee.

The Selection of Instructional Materials

The responsibility for selecting noncomputer instructional materials is usually shared by many individuals, including classroom teachers, librarians, media specialists, administrators, students, parents, and community members. No one person or group has total control over the choice of instructional materials. In fact, there is no uniform selection process or set of evaluation criteria in use in the United States. The most widespread approach for selecting instructional materials is the adoption process, and it is most frequently used to select printed instructional materials.

About half of the U.S. states have formal adoption procedures. This means that several textbooks are submitted to a committee for discussion and review. After a vote, textbooks are listed as "approved" educational material; this allows local schools within the state to use state-appropriated money to purchase these books.

Committees composed of consultants, curriculum coordinators, media specialists, and department chairpersons are most frequently responsible for selecting microcomputer materials. Past practice, the need to maximize limited budgets, and the scarcity of educators who are knowledgeable about microcomputers are a few of the reasons why schools tend to use a centralized approach to the purchase of microcomputer materials.

What Teachers Know about Selecting Materials

For many teachers, selecting instructional materials is largely an intuitive task. EPIE and other investigators have determined that the average classroom teacher has not been trained to evaluate or select instructional materials. Some experts believe that teachers are similarly unprepared to select and evaluate educational software.

> Many educational software developers are clearly skeptical about whether teachers would appreciate quality if they could get it. They believe that teachers will always opt for familiar and traditional educational materials, even if offered a choice that includes far better products. (Olds, 1981, p. 20.)

This pessimistic statement is supported by data about how schools use microcomputers. Drill-and-practice programs presently dominate the school software market because they are easy to create and resemble traditional printed instructional materials. By continuing to purchase programs that resemble printed materials converted for microcomputer use, untrained educators unwittingly remove the incentive to develop more sophisticated educational software.

How teachers locate, evaluate, and implement educational software is directly related to the problems of training teachers to use the new technology. Educators across the country face important decisions regarding the purchase and implementation of microcomputers. Schools often purchase microcomputer equipment without realizing fully the amount of training that will be needed for educators to use the equipment properly.

MICROCOMPUTERS AND TEACHER EDUCATION

The Need for Training

Educators are sometimes criticized for simply reacting to events rather than anticipating and planning for them. This complaint is valid in regard to the use of microcomputers in education. For years, experts warned that courses should be created to prepare teachers for the coming technological revolution. Now that the revolution has taken place and the race to integrate technology into education is underway, many teacher-training institutions and school districts are still trying to find the starting line.

Realizing that microcomputers have the potential to improve the quality of education, parents and other concerned citizens have prodded educators to catch up with the technology. In some instances, their efforts have created more problems than solutions.

> Under pressure from parents and school board members to begin computer instruction, "schools are behaving in exactly the wrong way," contends Phillip Mackey, an official of the School Boards Association in New Jersey, a state where many local districts have invested heavily in computers. "First they buy the machines. Then they buy the software," Mr. Mackey says. "Then they start to think, 'Why did we get into this in the first place?' " (Emmett, 1983, pp. 97–98.)

The training problems include where teachers will be trained to use microcomputers, who will train them, and what the training will include. Experts in the field do not agree on how to solve these problems; however, efforts to provide training to teachers are underway throughout the United States.

Sources of Teacher Training

Teachers are receiving microcomputer training from a multitude of sources, including local computer stores, computer books and periodicals, and professional organizations and conferences. School districts have formed cadres of trainers, offered evening classes, and allowed teachers to take computers home from school during vacations.

A number of teacher-training institutions are beginning to offer courses and degrees that focus on the educational uses of microcomputers. Mandates by state agencies have expanded the availability of courses to certify teachers in computer science and have provided inservice training for teachers in other disciplines.

The approach to training teachers to use microcomputers is frequently one of "inoculation": a quick dose of information, a little hands-on experience, and some encouragement. Unfortunately, continued support is all too often nonexistent.

What Teachers Must Know about Microcomputers

The philosophical arguments of educational theoreticians have not vanished with the arrival of microcomputer technology—they have reappeared as "old wine in new bottles." Some theorists believe that students need structured experiences with the computer, while others proclaim the benefits of an exploratory approach. It is highly unlikely that any one approach to instruction will ever be accepted or proven to be applicable under all conditions.

It is important to remember that the widespread use of computers in education is so recent that approaches for training educators in how to implement this new discipline are just emerging. A recommended strategy for all educators is to strive for a balanced perspective—one that comes from training in computer operation, programming, equipment, and instructional considerations.

OTHER FACTORS RELATED TO EDUCATIONAL SOFTWARE

The Home Computer Factor

Many factors affect the development of educational software. For example, because the home market for educational software is far greater than the school market, educational software publishers may unintentionally create a conflict among schools, parents, and students.

Advertisers create the impression that a child's future success or failure is directly linked to whether or not his/her parents purchase a microcomputer and educational software. This widespread use of microcomputers in the home may encourage parents to become quasi-curriculum coordinators. Parents may unknowingly select educational software that confuses their children or presents instruction that is not consistent with methods that the children are learning at school.

Machine Incompatability

The number of software programs available to educators is increasing daily; however, not all educators have access to the same universe of software. In most cases, educational software is "machine specific." This means that software written for a particular computer (and sometimes a particular model) can only be used with that brand.

Consequently, publishers tend to focus on producing educational software for those brands of computers that are regarded as the most popular for educational use. Presently, the majority of educational software is written for Apple, Commodore, IBM, and Radio Shack microcomputers. Schools that own other brands have fewer programs from which to choose.

Who Will Produce Educational Software?

The production of educational software requires resources that are typically not available to the small-scale developer. This fact has contributed to the decline of software production as a cottage industry.

Large publishing companies are entering the educational software business, motivated not only by potential sales but also by a concern that educational software might replace traditional printed materials. Already, there has been a dramatic increase in software geared to specific textbooks, and "textware" eventually will be the rule rather than the exception.

Internal Factors

Several states are attempting to influence the quality of educational software by inviting software publishers to particpate in meetings with educators. New York State's Center for Learning Technologies has sponsored software colloquiums during which educators and publishers discuss software features that educators have found to be problem areas. In addition, they often are shown videotapes of teachers and students using programs in a classroom setting. The success of these efforts is difficult to measure, but educators would do well to remember that they, as a group, have the power to affect the quality of software by refusing to purchase substandard materials.

External Factors

It seems that a degree of anguish has developed over the quality of American education, and the 1980s have spawned dozens of reports containing recommendations for improving public education. A 1983 report by the National Commission on Excellence in Education entitled "A Nation at Risk" drew much attention to issues in public education and is one report among many containing recommendations for improving teacher education and for incorporating microcomputer technology into American schools. Underlying many of these reports is a perception that the United States is losing its position as a leader in technology. The result is a new direction for public education in the 1980s. Funding for teacher training, research, and materials development in microcomputer technology are increasingly being made available to public education. These developments, along with new technological advances, will go a long way toward solving some existing concerns about educational software.

CONCLUSION

The microcomputer has presented many challenges to the public, publishers, and educators. The development and purchase of educational software present new and different issues from those posed by printed instructional materials. Learning to "pilot" or use the microcomputer for instructional purposes is still a relatively new area; thus, educators need to clarify the meaning of computer literacy as it applies to training teachers to use microcomputers. In creating courses, the emphasis should be on using the computer as a tool, rather than on understanding the technical details of how a computer operates. If these things can be accomplished, microcomputer technology can truly be a force of positive change in public education.

READINGS AND RESOURCES

Becker, Jay. *Microcomputers in the Classroom: Dreams and Realities.* International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403, n.d.

This 64-page booklet gives a concise summary of the problems and possibilities associated with using microcomputers in the classroom. It contains a number of interesting statistics and references.

Gall, Meredith. *Handbook for Evaluating and Selecting Curriculum Materials.* Newton, MA: Allyn and Bacon, Inc., 1981.

This is an excellent book for educators who are interested in learning more about the process of evaluating and selecting curriculum materials. It does not discuss educational software per se; however, the text is extremely readable and provides information pertinent to any form of curriculum materials.

Woodbury, Marda. *Selecting Materials for Instruction.* Libraries Unlimited, P.O. Box 263, Littleton, CO 80160, 1979.

This book was written at the beginning of the microcomputer era in the schools and provides a wealth of information about issues related to non-computer-based curriculum materials. Woodbury cites much of the existing literature on materials selection and provides an excellent perspective on factors to be considered when selecting materials.

REFERENCES

"Alpha, Beta, LVR." *EPIEgram* 13 (1984):1–7.

Becker, H. J. "Survey Maps Computer Use." *Electronic Learning* 5 (1986):12–13.

Becker, H. J. *Microcomputers in the Classroom: Dreams and Realities.* Eugene, OR: International Council for Computers in Education, 1982.

Emmett, A. "American Education: The Dead End of the 80s." *Personal Computing* 7 (1983):96–105.

Kamins, S., and Waite, M. *Apple Backpack.* Peterborough, NH: BYTE Books, 1982.

Kolata, G. "Playing Hardball with Software." *Science 83* 4 (1983):67–69.
"Microcomputer Software: Are Schools Getting What They Deserve?" *MICROgram* 1 (1982):2–3.
"Public Favors Computer Education Mandate." *Classroom Computer Learning* 4 (1984):18.
Woodbury, M. *Selecting Materials for Instruction*. Littleton, CO: Libraries Unlimited, Inc., 1979.

PROJECTS AND QUESTIONS

1. Examine several microcomputer programs in your subject area and make a list of recommendations about how you would modify each program. Do you feel you have the programming skills needed to alter any of the programs?
2. Conduct a survey of schools in your area to determine how many minutes per week students at each grade level use microcomputers. Compare your results with the Johns Hopkins study reported in this chapter. Explain the reasons for the results you obtain.
3. Write a letter to an educational software publisher requesting information about the publisher's methods for field-testing and revising their programs. Make a class chart that shows how many publishers responded to letters and what their responses were.
4. Survey a group of teachers to determine how many of them have received formal training in evaluating instructional materials. How important do you think such training is?
5. Discuss methods that schools could use to prevent software from being illegally reproduced.
6. Read and summarize three articles that describe programs for training teachers how to use computers in the classroom.
7. Find out how many students in your local high school use a computer at home. Make a list of the software each

student uses most frequently at home. Discuss the potential problems that can arise when students have or don't have access to a computer at home.

8. Make a chart comparing the quantity of educational software available for three popular brands of microcomputers.
9. Make a list of factors that affect the development of educational software. Discuss those factors you think are likely to be changed by 1990.
10. Examine ten educational software catalogs and make a list of vendors that have a preview policy.

2

Overview of CAI Software

INTRODUCTION

> Since we cannot know all that is to be known of everything, we ought to know a little about everything.

This advice was offered by Blaise Pascal, a seventeenth-century French mathematician and is applicable to educators who are interested in making intelligent decisions about selecting and using educational software. This chapter is written with Pascal's advice in mind. It presents (1) an overview of the history of computers, (2) a nontechnical introduction to computer equipment, and (3) information about the origins, design, and research related to the effectiveness of computer assisted instruction (CAI).

A BRIEF HISTORY OF COMPUTERS

Computer history can be viewed in terms of progress toward making computers smarter, faster, smaller, and cheaper. Since the development by the ancient Chinese of the abacus (the earliest known computational device), many intelligent individuals have dreamed of the possibility of machines that might be used to perform calculations more rapidly.

Blaise Pascal was the first person to develop an actual working model of an adding machine. His "machine arithmetique," completed in 1647, consisted of a series of gear-driven counters similar to those in an automobile odometer. However, Pascal's invention never became widely used because no one was particularly interested in mass-producing a complex mechanical adding machine.

Charles Babbage, a British engineer born in 1791, attempted to build the first general-purpose computer, but the technology necessary for implementing Babbage's ideas simply did not exist in the 1800s.

Only in recent times did a general-purpose computing machine become a reality. An electromechanical machine called the MARK I was developed in 1944 by Howard Aiken at Harvard University. By current standards, the MARK I was quite slow, requiring several seconds to complete a single arithmetic operation. It was a massive device that stretched to over 51 feet in length and stood over 8 feet high.

ENIAC, the first electronic computer, was developed in 1946 by John Mauchly and J. Presper Eckert at the University of Pennsylvania. ENIAC cost half a million dollars, weighed 30 tons, and required floor space equal to the size of an average house.

In the early 1970s, a breakthrough occurred with the development of a computer on a microchip. A microchip is a miniature circuit printed on a wafer-size piece of silicon, which enables a computer to perform millions of calculations in seconds. This was the breakthrough that made computers affordable for the general public.

COMPUTER BASICS

Types of Computers

Modern computers come in three "sizes": mainframe, mini, and micro. In general, these sizes correspond to the amount of information and complexity of applications that a computer can process. Mainframe computers are physically quite large and require special environmental conditions; they can store millions of characters of information. (This paragraph contains 795 characters—including blank spaces.) Minicomputers are about the size of a large cabinet, are capable of storing a million or more characters of information, and operate under "normal" environmental conditions. Microcomputers (the computers most frequently used by educators) are about the size of a typewriter, can store thousands of characters of information, and are suitable for use in classroom environments.

The Microcomputer System

A microcomputer system is typically comprised of a computer, a monitor, and a disk drive. In many cases, a microcomputer system includes an external device (peripheral) such as a printer. In one sense, a microcomputer system forms a symbiotic relationship with software, which contains the programming instructions stored on a disk. The microcomputer and software depend on each other to perform specific tasks. Figure 2.1 illustrates the parts of a typical microcomputer system.

Software selection and equipment selection are directly related. Careful selection of an instructional program is useless if the program is not designed to be used with your microcomputer system. The following sections discuss elements of a computer system that influence the selection and use of instructional software.

Computer Memory

A characteristic that distinguishes one microcomputer from another is the amount of information that can be stored in its temporary memory. A computer's temporary memory is called random access memory or RAM. When a software program is "loaded" into a computer, it is being put into the computer's RAM.

A computer's RAM is measured in kilobytes. The letter K stands for kilobyte and is roughly equivalent to 1,000 characters of

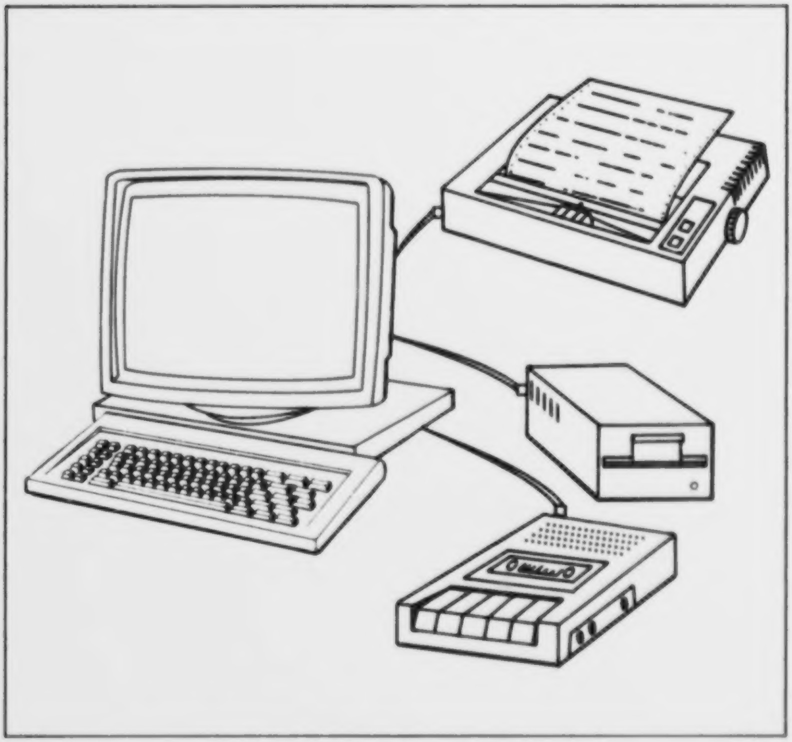


FIGURE 2.1

information. A 64K computer, for example, is capable of storing about 64,000 characters of information in its RAM. Likewise, a 48K computer can store about 48,000 characters in its RAM.

Classroom computers vary in RAM capacity depending on the model and brand. Computers with 64K to 128K of RAM are commonly used in many schools today. The RAM capacity of a computer must be considered when software is purchased because software programs have different memory requirements. A program that requires 128K of RAM will not operate on a computer with a 64K memory.

Cassette and Disk Storage

Programs can be stored on magnetic tape, cartridges, hard disks, cassette tape, or floppy disk. Each method has advantages and disadvantages. Cassettes can be loaded into a computer with an inexpensive cassette recorder. Two disadvantage of cassette-based systems are very slow loading speed (10 minutes for a 16K program) and limited storage capacity. The majority of programs written for classroom use are available on a floppy disk, or diskette, which is a thin piece of mylar inside a plastic jacket. These disks are usually 5¼ inches in diameter and vary in their capacity to store information. A typical single-sided, double-density 5¼ inch disk can store the equivalent of 55 double-spaced, typewritten pages.

Information is transferred between a disk and the computer via a disk drive, which is a mechanism that holds the diskette and causes it to rotate at about 300 revolutions per minute. Some disk drives are built directly into a computer while others are external to the computer. It is handy to have two disk drives for copying disks, and two disk drives are necessary for use with some programs; however, most instructional software is designed for use with one disk drive.

Monitors

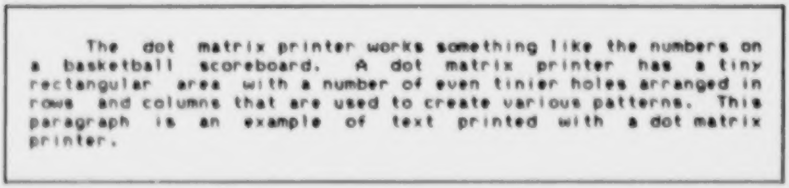
A cathode ray tube (CRT), the type of tube used in a television set, is used to display information, or output, from the computer. CRT displays (often called monitors) provide a picture that is much clearer than a standard televison receiver since word processing and most instructional programs demand a high-quality monitor.

Two types of monitors are available for use with computers: color monitors and monochrome monitors. Monochrome monitors, which vary in display size and background, are quite suitable for most educational software, but a number of programs require the use of a color monitor. It is important to have a color monitor to display a program whose content depends on color to convey an idea; for example, many programs for primary grade students use color to direct students' attention to specific areas of the screen. It is now possible to purchase high-quality color monitors that also clearly display word processing text in monochrome. In an ideal world, teachers would have access to both monochrome and color monitors so that they could match software type to display monitor.

Printers

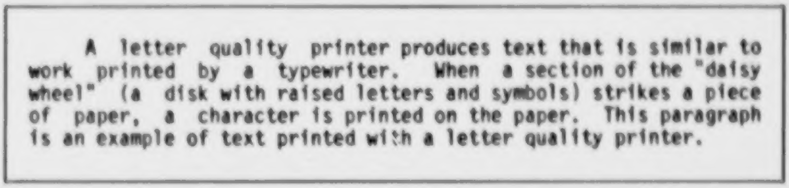
Some instructional programs require the use of a printer. One type of printer commonly used by educators is the "dot matrix" printer. As their name suggests, dot matrix printers construct each character in a series of dots within a matrix. They can print both text and graphics quite rapidly.

Another type of printer is the "letter quality" printer, which produces print similar in quality to a conventional typewriter by printing an entire character with a single strike; however, letter quality printers are generally slower than dot matrix printers. Figure 2.2 provides samples of text from both types of printers.



The dot matrix printer works something like the numbers on a basketball scoreboard. A dot matrix printer has a tiny rectangular area with a number of even tinier holes arranged in rows and columns that are used to create various patterns. This paragraph is an example of text printed with a dot matrix printer.

FIGURE 2.2a



A letter quality printer produces text that is similar to work printed by a typewriter. When a section of the "daisy wheel" (a disk with raised letters and symbols) strikes a piece of paper, a character is printed on the paper. This paragraph is an example of text printed with a letter quality printer.

FIGURE 2.2b

Equipment Selection

The task of selecting computer equipment for use in a school can be perplexing. The actual selection of computer hardware should be based on knowledge of the existing and anticipated software available for each brand of computer. It should also take into account such factors as personnel, school district purchasing policies, and available funds.

COMPUTER ASSISTED INSTRUCTION

The most common instructional use of microcomputers is computer assisted instruction (CAI), which is usually subdivided into the following modes: (1) drill and practice, (2) tutorial, (3) simulation, (4) gaming and problem solving.

CAI was first used in education in the late 1950s and early 1960s when universities began utilizing large mainframe computers. The TICCIT and PLATO projects are noteworthy examples of computer-based instruction for large computers.

TICCIT (Time-shared Interactive Computer-Controlled Instructional Television) was developed at Brigham Young University in the late 1960s and early 1970s. The system, one of the first to use color televisions coupled with a computer, is capable of providing instruction to 128 users simultaneously. TICCIT has not enjoyed very wide use, but its software is of above-average quality and is frequently referred to in CAI literature.

PLATO (Programmed Learning for Automatic Teaching Operations) was developed at about the same time at the University of Illinois. The PLATO system can support a large number of simultaneous users (as many as 2,000) and has an extensive library of instructional programs, some of which are now beginning to appear as software for microcomputers.

DRILL AND PRACTICE

A Definition of Drill and Practice

Drill-and-practice programs do not teach; they simply allow students to practice skills they have already been taught. Literal information such as arithmetic facts or vocabulary definitions are typical examples of content used in drill-and-practice programs. The best of the programs provide well-organized, self-correcting drill.

Historical Background

Drill and practice has been a part of education for centuries. After students receive instruction, they often need to practice what they've learned. Oral recitation, rote memorization, written tasks, and homework have traditionally been used for this purpose.

Drill-and-practice programs for computers have existed since the beginning of computer-based instruction. In 1963 at Stanford, Patrick Suppes and Richard Atkinson began a program of research and development in drill-and-practice CAI, and their early work was concerned with the development of materials in language arts and mathematics for disadvantaged elementary school students.

The results of a 1984 software survey conducted by Educational Products Information Exchange showed that the majority of educational software on the market consisted of elementary school drill-and-practice programs (see Figure 2.3).

Mapping the Software Territory

Using over 5,000 educational software programs listed in TESS, *The Educational Software Selector*, Educational Products and Information Exchange (EPIE) compiled the list of software shown below. The list indicates that drill-and-practice software dominates the software market, with tutorial programs a distant second, and games third. Because the categories listed below overlap, the totals will exceed 100 percent.

Software by Subject	
Mathematics	25.8%
Science	13.5
English/Language Arts	12.0
Reading	9.9
Social Studies	5.2
Computers	5.1
Foreign Languages	4.5
Other (20 additional categories)	24.0

Program Types	
Drill and practice	69.8%
Tutorial	31.8
Game	20.0
Computational Support	8.7
Simulation	8.1
Teacher Aid	3.8
Other	9.4

Grade Levels	
Preschool/Kindergarten	10.0%
Primary	35.4
Intermediate	52.9
Junior High	50.9
Senior High	60.9
College	31.1

From *EPIEgram*, 1985, p. 4

FIGURE 2.3

Common Drill-and-Practice Designs

Drill-and-practice programs present the learner with a problem to be solved or a question to be answered. Some common response formats for drill-and-practice programs are presented below.

- *A two-way-choice alternative*

```
The great Wall is located in China.

Type Y or N.
```

- *Multiple-choice items*

```
One gallon is equal to ____________

A. 4 pints
B. 4 liters
C. 4 quarts
D. 4 ounces

Type the letter of the correct answer.
```

- *Fill-in-the-blank items*

```
6 × 5 = ____

Type the correct answer.
```

These response formats are frequently used in printed instructional materials. When applied to computerized drill-and-practice programs, they may be creatively presented to remediate and reinforce important skills; however, in many instances, drill-and-practice programs simply resemble workbooks and fail to use the unique capabilities of the computer.

Examples of Drill-and-Practice Programs

Figure 2.4 shows a screen display from *Alien Addition,* a drill-and-practice program published by Developmental Learning Materials (DLM). This program uses the sound, color, and graphic effects presented in popular arcade games as a basis for motivating students to memorize basic arithmetic facts. The object of *Alien Addition* is to answer a problem correctly in order to destroy an alien spaceship. The microcomputer continuously presents problems to the student at timed intervals.

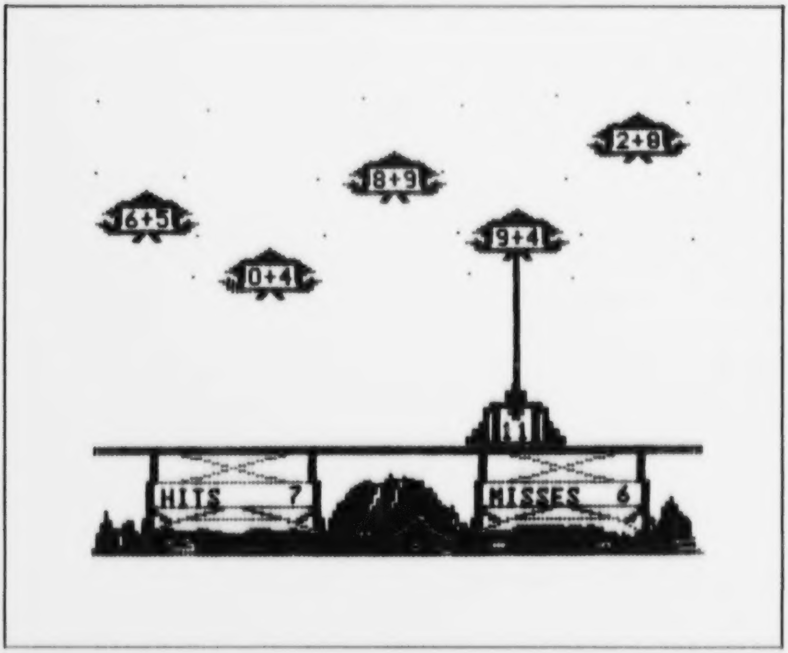
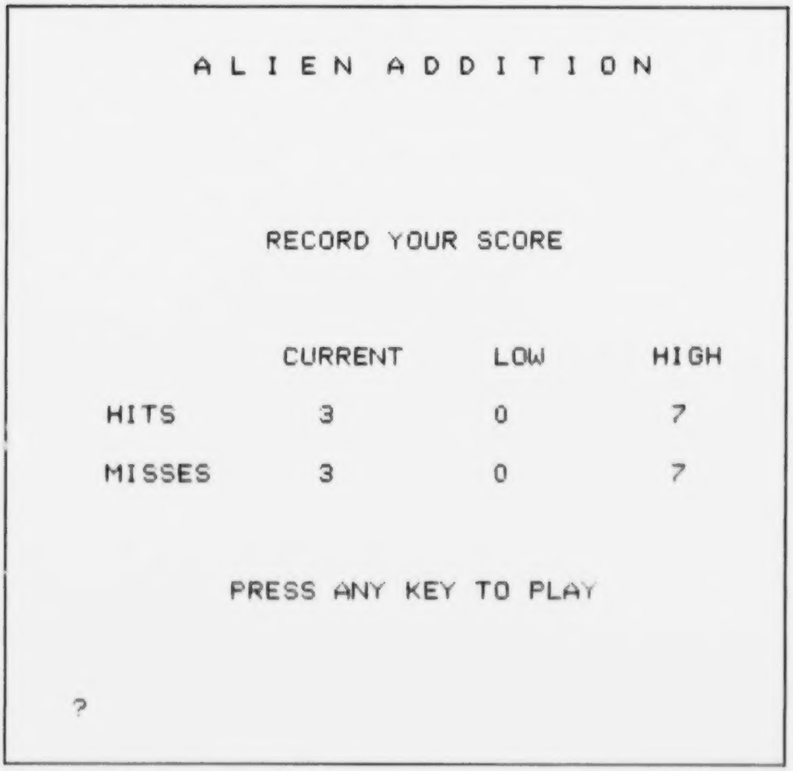


FIGURE 2.4

© DLM Teaching Resources, Allen TX 75002.

The program automatically records a student's score, reports the results in an end-of-program summary, and includes printed practice materials for students to use off-computer (see Figure 2.5).

Figure 2.6 shows a screen display from *Music Theory*, a program produced by the Minnesota Educational Computing Corporation (MECC). This program provides students with drill and practice in various music skills. The screen display pictured is from a drill called "Wrong Note," which presents the learner with a picture of a phrase of music. The student's task is to listen to the phrase when it is played by the computer and select the wrong note. The program corrects the student's errors by displaying the correct phrase and replaying the music.

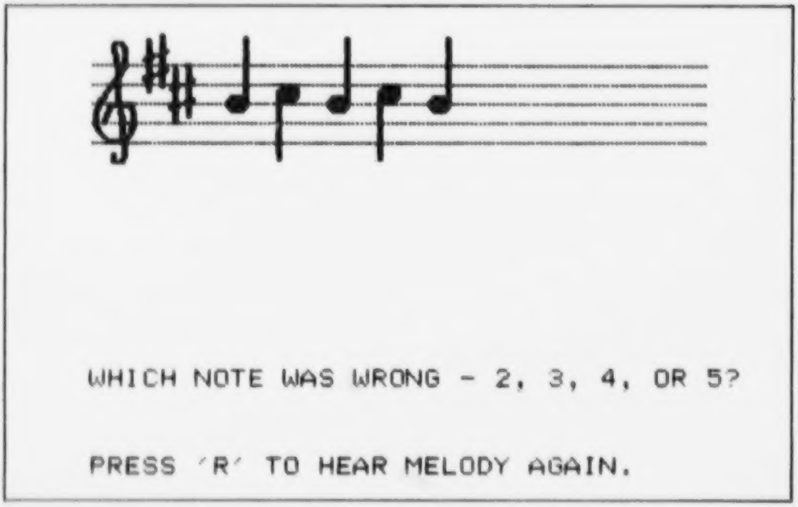


```
              A L I E N   A D D I T I O N



                  RECORD YOUR SCORE


                   CURRENT      LOW       HIGH

      HITS           3           0          7

      MISSES         3           0          7


                PRESS ANY KEY TO PLAY



  ?
```

FIGURE 2.5

Criticisms of Drill and Practice

A number of critical comments have been directed at drill-and-practice programs. A frequent criticism is that microcomputer resources are too limited and too expensive to be used as automatic flash card machines. Poorly conceived drill-and-practice programs have been characterized as monotonous and at times have been criticized for reinforcing incorrect learning. Another concern is that drill-and-practice programs may negatively influence a student's perception of instruction by reducing learning to a trivia game that implies that all questions have only one correct answer.



WHICH NOTE WAS WRONG - 2, 3, 4, OR 5?

PRESS 'R' TO HEAR MELODY AGAIN.

FIGURE 2.6

This has been taken from *Music Theory*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.

The Potential Value of Drill and Practice

Are drill-and-practice programs the "one-celled organisms" of CAI, totally without merit, and a poor use of a powerful microcomputer? Most educators acknowledge that every content area has a domain of important facts that students must learn. A modern view of the learning process accepts drill and practice as a necessary part of that process. Studies of conventional teacher-directed instruction report that the frequency of factual single-answer questions is positively related to gain in student achievement. Stallings and Kaskowitz (1974) found that the frequency of academically focused, direct questions at the lower levels of Bloom's taxonomy of educational objectives resulted in increased acquisition of basic skills in arithmetic and reading.

Using microcomputers to provide students with drill and practice can eliminate the potential drudgery of repetitious tasks for both teachers and students. Drill and practice has traditionally been associated with print materials and the presence of a teacher;

however, requiring a teacher to prepare, present, and monitor practice tasks for students limits the teacher's opportunity to teach. Microcomputers are well-suited to presenting repetitious tasks and patiently responding to students.

Research on Effectiveness

CAI drill programs have been the subject of a number of studies. Research indicates that computer-based drill-and-practice can be more effective than traditional instruction. Becker (1982) reported that "major, systematic, well-monitored evaluations have found that computer-based drills can produce increased achievement in comparison to prior years or in comparison to control group treatments" (p.19). These results have been found most often in mathematics and foreign language instruction. Forman (1982) also reported that drill-and-practice programs seem to be more effective for low-ability students than for middle- or high-ability students.

TUTORIAL CAI

A Definition of a Tutorial

In contrast to drill-and-practice programs, tutorial CAI usually attempts to introduce new information and concepts. As the name implies, a tutorial is designed to tutor the user.

Tutorial programs often use instructional design principles originally developed for programmed instruction. The term "instructional design" was developed by learning theorists to describe a model-based, systematic approach to creating educational materials. Programmed instruction (PI) refers to a particular instructional design for presenting printed learning material, and its common characteristics include (1) an ordered sequence of instruction, (2) the presentation of information in small increments, (3) active student responses, (4) a narrow range of possible answers, and (5) provisions to reinforce correct responses while informing the learner about his/her progress.

Historical Background

During the 1950s and early 1960s, educational researchers used programmed instruction techniques with machines that were designed to automate the presentation of frames containing information (see Figure 2.7). These "teaching machines" were based on the ideas of B.F.Skinner. Practical experience and research soon indicated that teaching machines were cumbersome and expensive page-turning machines.

Common Tutorial Designs

Studies of printed programmed materials have shown that some students consider programmed instruction tedious and boring. These programs typically use a linear instructional design, which requires a learner to progress from one point of instruction to another without deviating from the linear path. For a student to end an instructional unit or module in a linear tutorial program, he or she must successfully complete each task in sequence. A linear design is illustrated below.

```
Begin——Task——1——Task——2——Task——3——End
Instruction                       Instruction
```

While many tutorial programs for microcomputers use a linear design, an alternative design is a program that branches. Programs that use a branching design are laid out as a decision tree where the path taken by the student depends upon his or her response to questions at specific branch points. A branching

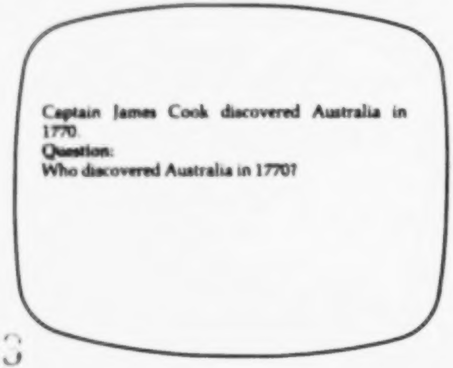
FIGURE 2.7

Frames and Pages ...

Robert Burke has defined a **frame** or **page** as follows:

> ...**frame** is a conventional term that has come to mean the amount of information that is presented on the microcomputer display screen at any one time. A frame may contain straight narrative text or it may contain a question, or a combination of both. The concept of a frame is useful as an indexing tool—a way of referring to a specific part of a CAI lesson or as one unit of measure of lesson size...Some CAI authoring systems have their own terminology and use **page** as an equivalent term. I prefer frame. (p. 68)

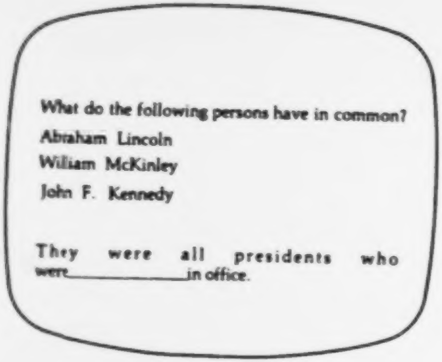
Frames will vary in quality and purpose. The following are three examples of "frames" or "pages":



Captain James Cook discovered Australia in 1770.
Question:
Who discovered Australia in 1770?

The copy **frame**. In this frame the response is copied from the material in the frame. This is regarded as the lowest form of frame life and is often associated with the worst types of CAI.

The Egrule frame. Egrule frames start with carefully selected examples and require the student to deduce relationships and rules (adapted from Burke, 1983).



What do the following persons have in common?

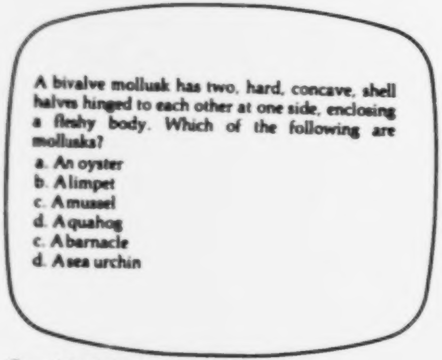
Abraham Lincoln

William McKinley

John F. Kennedy

They were all presidents who were__________in office.

The Ruleg frame. The ruleg frame is the inverse of the egrule frame. A rule is provided first and the student is then required to apply the rule to examples (adapted from Burke, 1982).

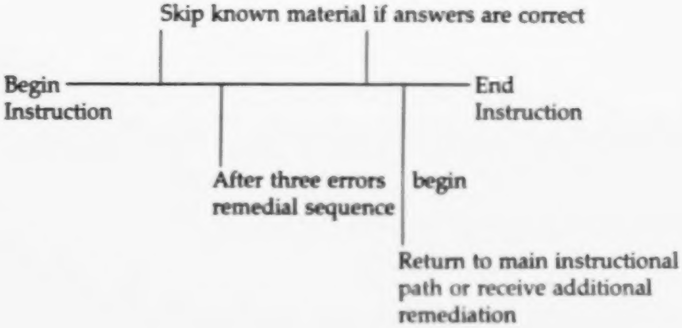


A bivalve mollusk has two, hard, concave, shell halves hinged to each other at one side, enclosing a fleshy body. Which of the following are mollusks?

a. An oyster
b. A limpet
c. A mussel
d. A quahog
c. A barnacle
d. A sea urchin

From *Microcomputer Applications in the Classroom* by Alan Hofmeister, Copyright © 1984 by CBS College Publishing. Reprinted by permission of CBS College Publishing.

program allows the student who correctly answers a question to skip ahead to new information, while the student who gives an incorrect answer is routed to a sequence of remedial instruction. The pattern of instruction in a branching program resembles the branches of a tree and is illustrated below.


Skip known material if answers are correct
Begin Instruction
End Instruction
After three errors begin remedial sequence
Return to main instructional path or receive additional remediation

Most tutorial programs interact with students on a limited basis. The programs present information and questions, and the student provides a response. Tutorial programs are usually designed to accept a fairly narrow range of responses; however, more sophisticated programs have been developed for mainframe computers and are able to provide an interactive dialogue with the learner based on the learner's response to specific questions. An example of a dialogue between a computer (C) and a student (S) is shown on page 37.

C: Hello,
What's your name?

S: Mike.

C: Hello, Mike.
I'm going to teach you a rule about temperature.
The closer a country is to the equator the warmer its climate.
Look at the map in your textbook, Mike.
Is Brazil closer to the equator than Finland?

S: Yes.

C: That's correct, Mike.
So which country has the warmer climate?
Brazil or Finland?

S: Brazil.

C: You're correct again Mike.
Tell me why it's warmer.

S: Because it's closer to the equator.

C: Right.
Look at your map again.
Tell me the name of another country that's warmer than Finland.

S: Cincinnati.

C: That's not a country, Mike.
Please try again.
Tell me the name of a country that's warmer than Finland.

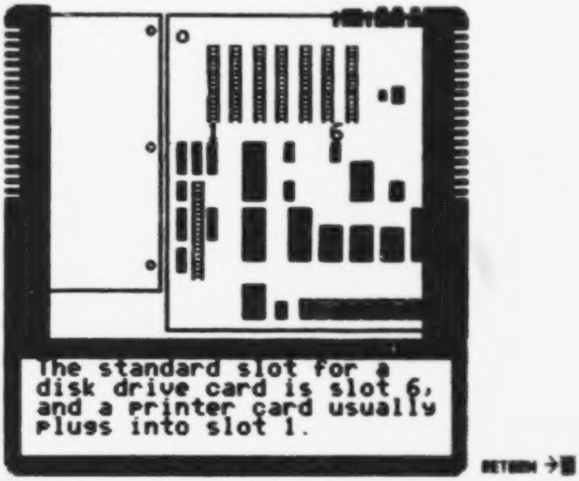
This type of tutorial is sometimes referred to as intelligent CAI (ICAI). These programs attempt to analyze the learner's understanding of material and provide instruction based on this analysis. A small number of ICAI programs are available primarily for use with mainframe computers. While such programs more closely resemble human teacher-pupil interaction, they are extremely difficult to design, so ICAI programs for microcomputers are not likely to become commonplace until suitable systems are available at affordable prices and a wider range of ICAI expertise has developed.

Examples of Tutorial Programs

Some excellent tutorial programs have been created to teach the use of computers or software programs. Figure 2.8 shows a screen display from *Know Your Apple IIe,* a tutorial program that teaches about the Apple IIe microcomputer.

Microtype is a step-by-step introduction to keyboarding on a microcomputer. The screen display shown in Figure 2.9 is an example of the instruction presented in the program.

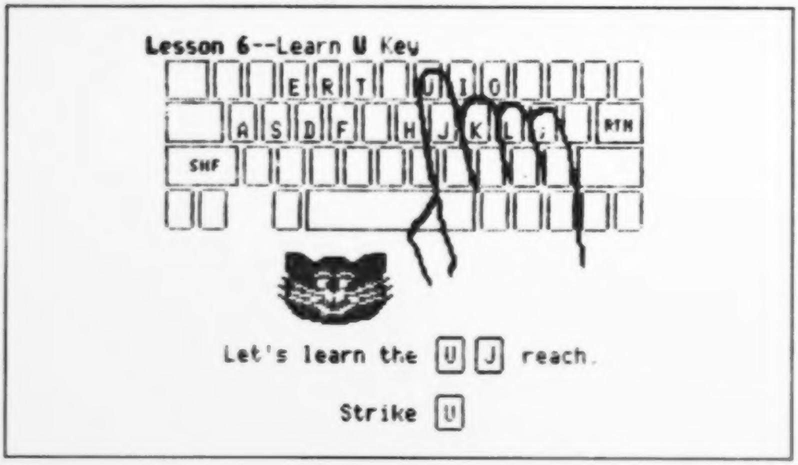
Figure 2.10 shows a screen display from the MECC tutorial program called *Writing a Character Sketch,* which presents students with examples and questions to think through in order to determine what is needed to characterize a person from fiction, history, or their own experience.



```
The standard slot for a
disk drive card is slot 6,
and a printer card usually
plugs into slot 1.
```

FIGURE 2.8

Courtesy of MUSE Software.



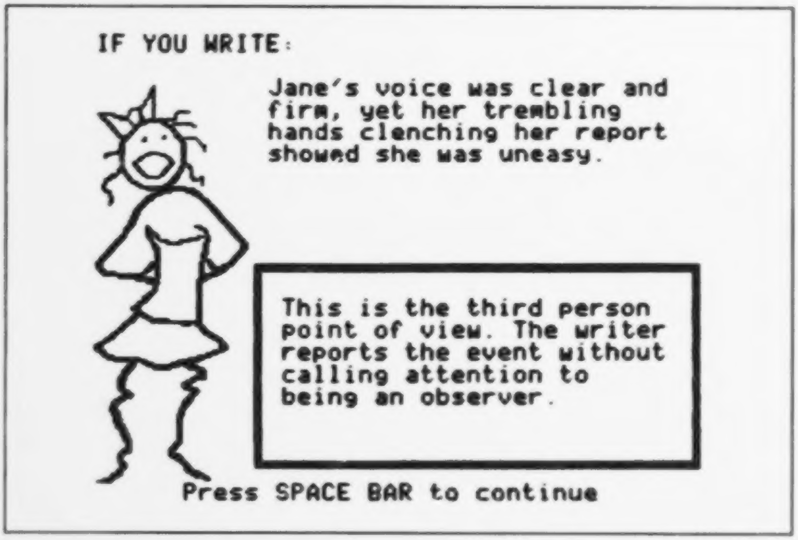
Lesson 6--Learn U Key

Let's learn the U J reach.

Strike U

FIGURE 2.9

Reprinted from *Microtype: The Wonderful World of Paws* by EduSystems, Inc., published by Southwestern Publishing Co.



IF YOU WRITE:

Jane's voice was clear and firm, yet her trembling hands clenching her report showed she was uneasy.

This is the third person point of view. The writer reports the event without calling attention to being an observer.

Press SPACE BAR to continue

FIGURE 2.10

This has been taken from *Writing a Character Sketch*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.

Limitations of Tutorials

Tutorials have several limitations that make the programming of precise instruction difficult. A tutorial designer must anticipate all possible ways that a learner might respond to a program. Deciding what constitutes a correct answer and how a tutorial will respond to errors requires extremely careful planning. Various types of possible student answers must be considered. Some examples are presented below.

- *Correct or synonymous answer*

```
C:  Name the third planet in the solar system.
S:  Earth.
C:  That's correct. Your answer matches with my answer.
```

The student's answer is a perfect match with the answer anticipated in the program.

- *Literally incorrect but conceptually correct answer*

```
C:  Name the third planet in the solar system.
S:  Erth.
C:  That's incorrect. Your answer does not match with my
    answer.
```

The student's answer is conceptually correct but is regarded as an error because it is misspelled. The program's design does not enable it to accept approximations of the correct spelling of a word.

- *Anticipated errors that reflect mislearning*

```
C:  Name the third planet in the solar system.
S:  Venus.
C:  That's incorrect. Venus is the second planet in the
    solar system.
```

The student's answer indicates that (a) the student may not have been taught that the sun is not a planet, thus affecting the student's ordering of the planets or (b) the student simply does not recall the correct answer.

- *Unanticipated, but not necessarily incorrect, answers*

```
C: Name the third planet in the solar system.
S: Planet Earth.
C: Planet Earth is not the name of a planet in the solar
   system.
```

The student's answer is correct; however, the program's design does not enable it to ignore the word "planet" and simply search for the word "Earth."

It is also quite difficult to design tutorial instruction for content that does not readily conform to rules or procedures. Consider the task of writing a paragraph with a main idea and appropriate supporting details. It is not possible to specify rules so that a computer can judge if a student has written a well-structured paragraph or nonsense.

In the conventional learning environment, the teacher can respond to students' questions and problems in a variety of ways, but a microcomputer tutorial is not able to ad-lib judgments because students rarely respond to instruction in completely predictable ways.

Research on Effectiveness

Research related to the effectiveness of tutorial CAI has often been concerned with the issue of instructional control. Should the rate, sequence, and content of instruction be prescribed by the student or by the tutorial program? The research indicates a compromise position on learner control: "Either the student or program may control the course of learning at appropriate times" (Kearsley, Hunter, and Seidel, 1983).

This information, paired with Forman's (1982, p. 38) observation that "the bulk of the studies showing CAI to be effective have concerned the use of adjunct CAI in which the classroom teacher was readily available," indicates that tutorial programs should be viewed as a resource in a teacher-directed classroom situation. Teachers do not relinquish their instructional responsibility when tutorial programs are used and the successful implementation of tutorial programs is predicated on the teacher's active participation. Two of the major responsibilities of a teacher who uses tutorials are (1) to evaluate and select programs that match the learner's needs and (2) to provide instruction and assistance to learners before, during, and after involvement with a tutorial program.

SIMULATION-ORIENTED CAI

A Definition of a Simulation

A simulation imitates an abstract or real-life situation or process. Participants in a simulation usually play a role that requires them to interact with other people or elements of the simulated environment. Simulations frequently use a game-like format to help produce a relaxed atmosphere that is conducive to learning.

Historical Background

Simulations have been used for years in the form of board games. Board games such as Monopoly and Life have been popular with home users, while instructional simulation games such as Planet Management and Empire have been used in high schools and colleges.

Computer simulations were first used by the military to teach combat tactics and strategies. Instructional simulations evolved from so-called "crisis games" that were developed during the 1950s and consist of hypothetical situations where participants try to solve a crisis within a framework of rules based on reality.

Instructional simulations became popular teaching tools during the 1960s and early 1970s. During that time, Jerome Bruner and other educational theorists advocated active student involvement in learning. Informed guessing and hypothesis testing were believed to be the best methods for promoting "discovery" learning. A number of computer-based simulations were developed during the 1960s and 1970s, including the Huntington Projects, which began in 1968 and produced computer simulations for biology and the social sciences that are now regarded as classics.

Common Simulation Designs

The design and programming of a microcomputer simulation is a complex task that requires hundreds of hours of planning and programming. Three designs are commonly used for educational simulations.

Procedure Simulations. Some instructional goals require students to become proficient at performing tasks that are potentially dangerous or expensive (for example, correctly diagnosing and prescribing treatment for an illness or landing an airplane). Procedure simulations allow students to practice these and other procedural skills without incurring tremendous expense or the loss of lives.

Information Retrieval Simulations. The goal of an information retrieval simulation is to obtain information. This type of simulation frequently has no clear task or performance goal for the learner other than to learn how to predict the consequences of some action. This simulation design has been used in programs that allow the learner to obtain information and test ideas about a simulated environment such as an economic or ecological system.

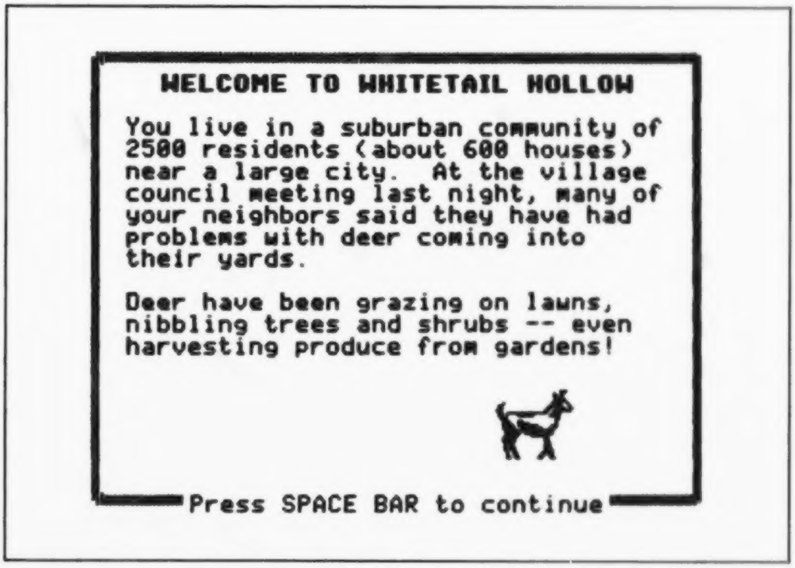
Encounter Simulations. Encounter simulations are concerned with phenomena that are unpredictable or difficult to define. The purpose of the simulation is to allow the user to experience achieving a goal in a setting that has no clear procedures or principles.

Examples of such programs are simulations of a teacher's first year in teaching, business competitions, or other role-playing activities.

An important attribute of many simulation designs is the opening scene or scenario which describes the objectives of the simulation, the goal of the participant, rules, time limitations, and related resources. Figures 2.11 through 2.13 are screen displays from the opening scenario of *Oh, Deer!*, a science simulation produced by MECC.

Examples of Simulation Programs

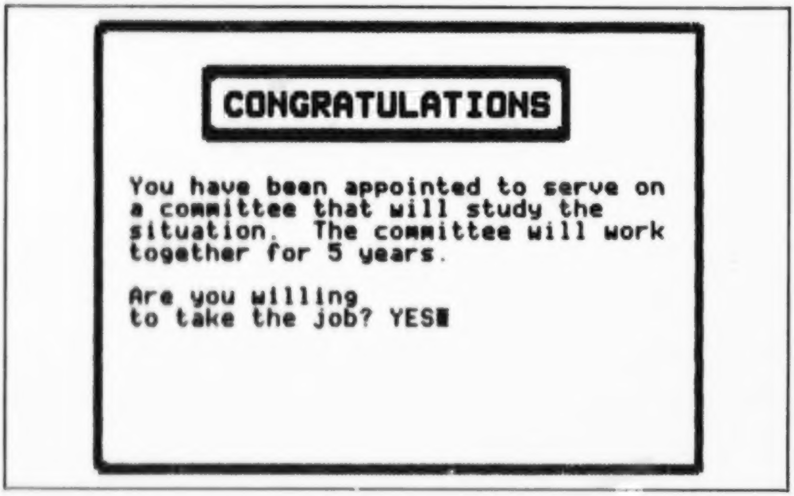
Figure 2.14 shows a screen display from another MECC simulation, *Discovery Lab*. The program allows the user to control and manipulate variables in a scientific experiment.



```
WELCOME TO WHITETAIL HOLLOW

You live in a suburban community of
2500 residents (about 600 houses)
near a large city.  At the village
council meeting last night, many of
your neighbors said they have had
problems with deer coming into
their yards.

Deer have been grazing on lawns,
nibbling trees and shrubs -- even
harvesting produce from gardens!

Press SPACE BAR to continue
```

FIGURE 2.11

This has been taken from *Oh, Deer!*, published by MECC,
3490 Lexington Avenue North, St. Paul, MN 55112.

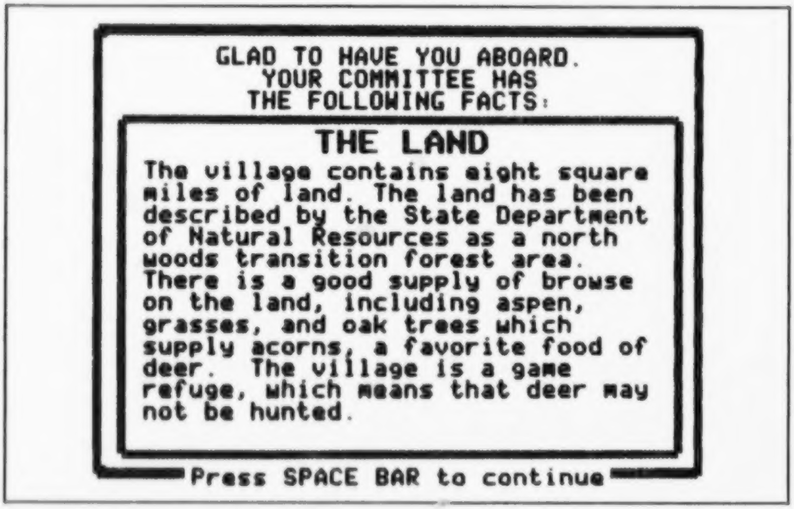


CONGRATULATIONS

```
You have been appointed to serve on
a committee that will study the
situation.  The committee will work
together for 5 years.

Are you willing
to take the job? YES█
```

FIGURE 2.12

This has been taken from *Oh, Deer!*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.



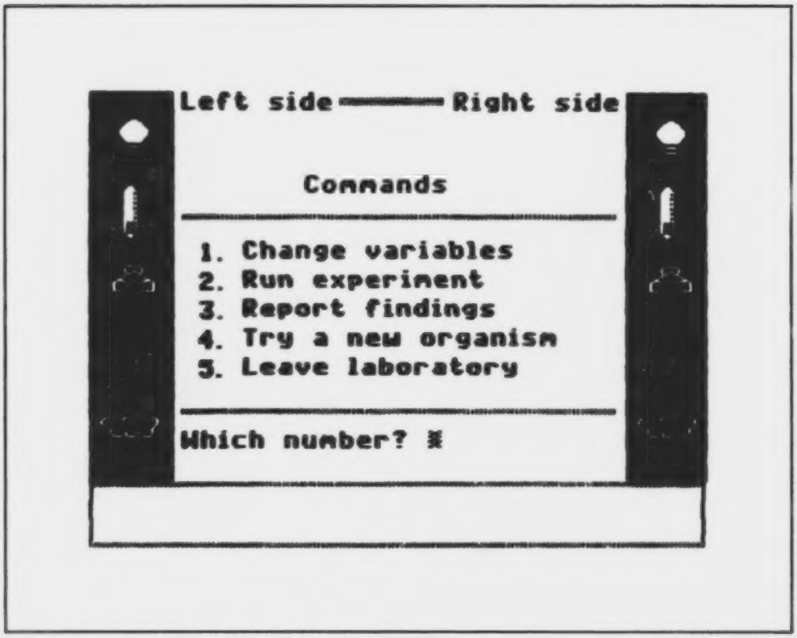
```
GLAD TO HAVE YOU ABOARD.
YOUR COMMITTEE HAS
THE FOLLOWING FACTS:
```

THE LAND

```
The village contains eight square
miles of land. The land has been
described by the State Department
of Natural Resources as a north
woods transition forest area.
There is a good supply of browse
on the land, including aspen,
grasses, and oak trees which
supply acorns, a favorite food of
deer.  The village is a game
refuge, which means that deer may
not be hunted.
```

FIGURE 2.13

This has been taken from *Oh, Deer!*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.



Left side —— Right side

Commands

1. Change variables
2. Run experiment
3. Report findings
4. Try a new organism
5. Leave laboratory

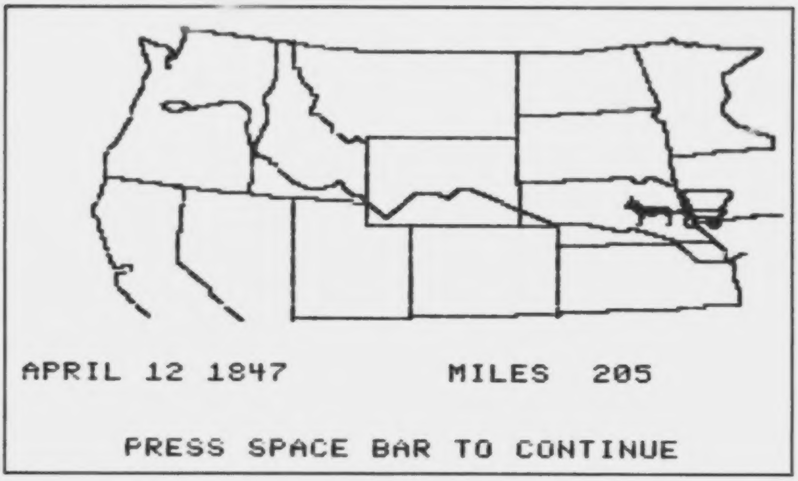
Which number? ▒

FIGURE 2.14

This has been taken from *Discovery Lab*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.

A social studies simulation program produced by MECC is *Oregon* (shown in Figure 2.15). This program allows students to simulate the experience of traveling the Oregon Trail from the Midwest to Oregon. Students must carefully allocate their "resources" to successfully complete the journey. It is an excellent example of a program that can be used to promote cooperative learning.

Rocky's Boots is a program that introduces concepts about electrical circuits. Students use this knowledge to direct the flow of electricity through a set of simulated machines. Figure 2.16 illustrates a screen display from a lesson about sensors and current flow. The program enables students to construct machines that require the application of information presented in the program.

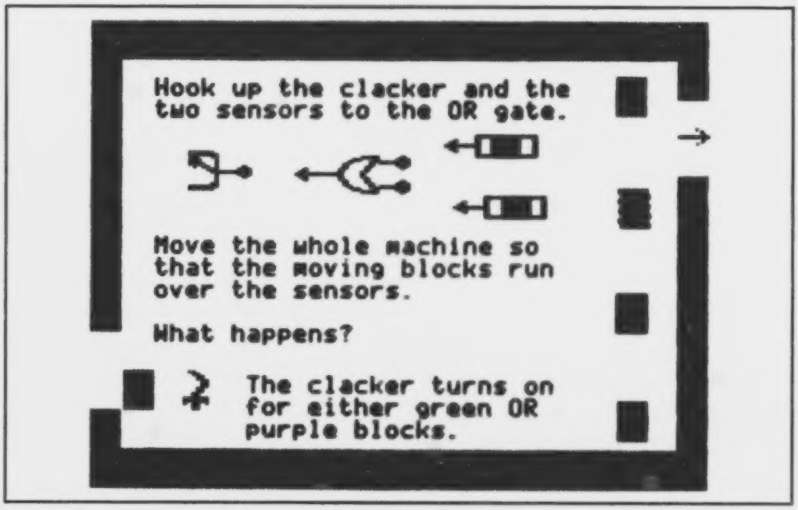


APRIL 12 1847 MILES 205

PRESS SPACE BAR TO CONTINUE

FIGURE 2.15

This has been taken from *Oregon*, published by MECC,
3490 Lexington Avenue North, St. Paul, MN 55112.



Hook up the clacker and the two sensors to the OR gate.

Move the whole machine so that the moving blocks run over the sensors.

What happens?

The clacker turns on for either green OR purple blocks.

FIGURE 2.16

The *Rocky's Boots* software program by The Learning Company.

Limitations of Simulations

Simulations can be used to study events that might not be examined in real life either because of expense, lack of time, or danger. Simulations are particularly valuable for instruction in science and social studies, areas in which events may be difficult to experience directly. As with all instructional materials, simulations have limitations as well as strengths.

One limitation of simulations is their dependence on the discovery method of learning, which involves creating conditions whereby learners can achieve insights on their own. Discovery learning via computer simulations frequently depends on the student's ability to read and interpret information. This can be a real handicap for students with limited reading skills.

The use of simulations frequently requires the grouping of students into pairs or small groups. While such grouping arrangements promote social interaction between students, some learners are not able to exercise the self-discipline and responsiblity necessary to be successful in self-directed instruction.

Dennis (1979, p.7) observed that "learning from a simulation is predicated on interaction with it. A simulation does not *present* information to the student. Rather it surrenders it as a consequence of student interaction." Because discovery learning is more time-consuming than teacher-directed instruction, simulations frequently require more opportunities for interaction than currently exist in many schools. An idea that might be stated in a single sentence could require an hour of interaction with a computer simulation. A teacher must decide if the simulation learning experience is worth the time.

Potential Value of Simulations

Determining the value of a simulation depends on the teacher's instructional goals. If the goal of instruction is to learn new skills as quickly as possible, then computer simulations are not the most effective vehicle. However, if the goals of instruction are to develop decision-making skills, personal interaction skills, and the ability to discover cause-and-effect relationships, then simulations can be very useful instructional tools.

Unlike most CAI programs, simulations are not limited to use by one student at a time. This is particularly benenficial for schools with a limited number of computers. It is possible to use many simulation programs with an entire classroom of students divided into decision-making teams that take turns using a single computer.

Research on Effectiveness

At the present time there is a limited amount of research about the instructional effectiveness of simulation programs. The research that has been done has been primarily concerned with noncomputer-based simulations. Greenblat and Duke (1981, p. 149) summarized the status of simulation research as follows:

> In summary, then, it appears that there is a considerable amount of anecdotal material about the affective involvement of students at the time of participation, and some evidence that student interest in simulations is very high. There is, at the moment, little hard data to show that such participation leads to greater interest in the subject matter, the course, or learning in general.

INSTRUCTIONAL GAMES

A Definition of Instructional Games

An instructional game is an activity in which players try to attain a goal within prescribed rules and may include elements of drill and practice, tutorial, and simulation CAI. Games can provide an attractive framework for learning activities and may provide students with a welcome break from the day-to-day classroom routine. Is there a difference between an instructional game and a recreational game? Some educators facetiously reply that the difference is that a game is fun while an instructional game is not!

Historical Background

Games are a form of play, and as such they are regarded by some educators as being incompatible with instruction. However, developmental psychologists such as Jean Piaget view games as natural learning activities. This viewpoint has encouraged many educators to use such games as Bingo, Tic-Tac-Toe, Scrabble, and others in the classroom, and almost every textbook series includes instructional games.

A number of computer games were originally developed for mainframe computers; however, instructional computer games for microcomputers are more closely modeled after video games designed for the general consumer. A trip to any computer store or local video arcade will demonstrate the popularity of computerized video games (computer games primarily used in the home with a television set and a computer designed for game playing). Like popular movies or songs, Space Invaders, Pac Man, and other video game titles will be remembered long after their popularity has waned.

The popularity of video games has created concern among parents and educators. Their concerns center on the amount of money that children spend to play the games, the games' violent themes, and the limited intellectual content of most video games. Some communities have banned video game arcades or tried to discourage the opening of new arcades by imposing a heavy gaming tax on arcade owners. It is interesting to note that many of the most ardent video game players are not children but adults. The debate over video games continues between educators who believe the games have no value and those who believe they do.

Common Instructional Game Designs

Various approaches are used in designing instructional games for the computer. While many computer games are simply traditional games (such as Hangman or Tic-Tac-Toe) transferred to a computer medium, others have an instructional objective "embedded" in their design. Because many instructional games are designed to be fun for students, it is often difficult to detect the instructional component.

There are many different types of game designs. David Ahl (1973) has categorized computer games into twelve categories: (1) guessing, (2) piles of objects, (3) matrix, (4) history/government, (5) sports, (6) logic/strategy, (7) gambling, (8) card and board, (9) war, (10) space, (11) word, and (12) data. The games described by Ahl often include the following elements:

Goals. A challenge presented to the learner becomes the focus for playing the game. Challenges are often tied to accumulating points, shooting targets, or completing a task within a specified time period.

Scorekeeping. Scorekeeping is frequently used as a motivational device. As a student progresses toward achieving the goal of a game, his/her score improves. A player's sense of continuing improvement is probably the most important motivating element in a game.

Adjustable Skill Level. Instructional games frequently allow the learner or the teacher to adjust the level of difficulty in a game so that the learner can achieve some measure of early success.

Competition. Instructional computer games usually require the learner to compete against another player, against the computer itself, or against the clock.

Chance. Many computer games involve elements of chance. A lack of predictability tends to pique the curiosity of students and keep them alert.

Audio and Visual Effects. Instructional games usually incorporate graphics that are tied to the theme of the game. Sound effects sometimes provide a player with feedback about his/her success, but they are usually used to make the game appear more attractive.

Examples of Instructional Games

Mastertype is an instructional game that presents information about touch typing through a sequence of activities. The program drills the player in various typing tasks using an arcade game format. The student is a planetary defender fighting the deadly invaders from the planet Lexicon. The invaders are depicted as letters or words in each corner of the screen. The words launch missiles at the defender's planet, which is in the middle of the screen (see Figure 2.17). The defender can destroy the approaching missiles by typing the correct letters or words on the computer's keyboard.

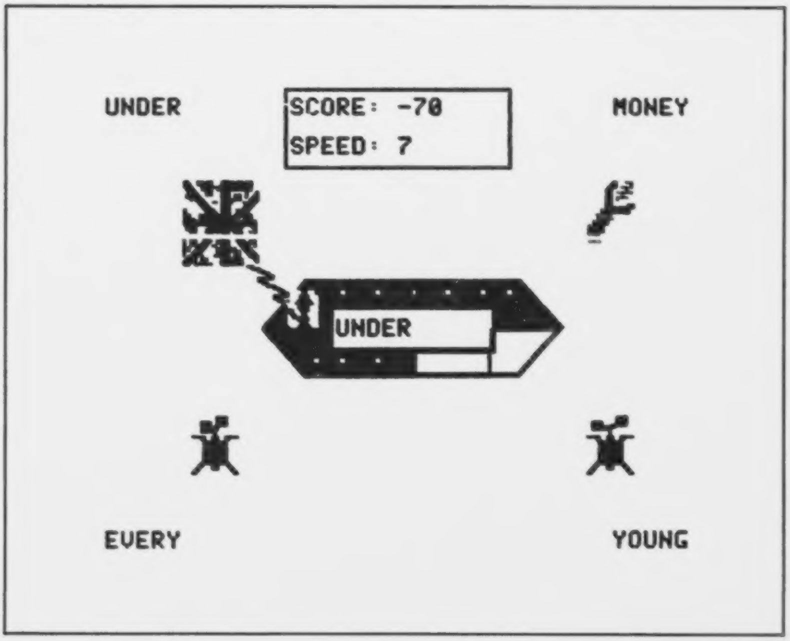
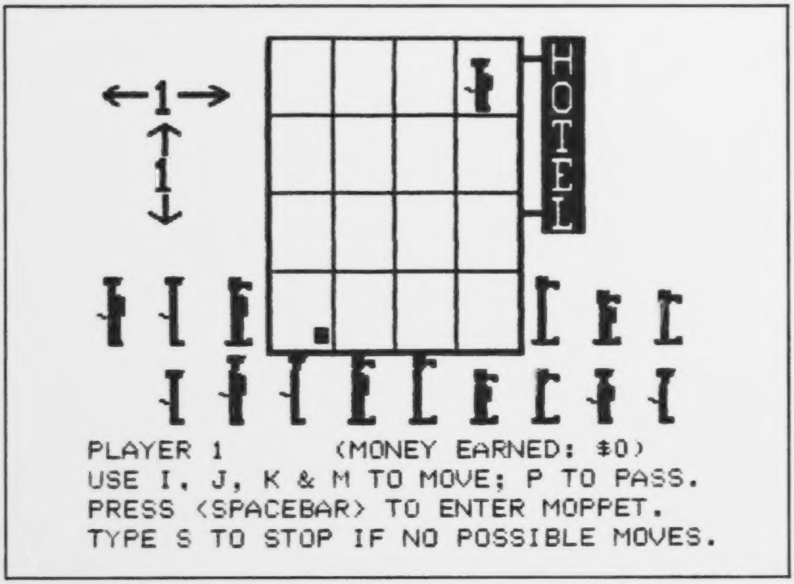


FIGURE 2.17

© 1983–1985 Scarborough Systems, Inc., Tarrytown, NY.

Moptown Hotel presents a series of games designed to teach elementary school students attribute recognition. Figure 2.18 is a screen display from *Moptown Hotel*. Acting as a hotel clerk, each player tries to put the most "moppet" characters into the hotel while following the rules presented on the screen. The format of the game requires students to plan strategically and to use a variety of logic skills.

The goal of *Number Farm* is to offer meaningful computer learning experiences for preschool or primary-grade children. The program presents simple games designed to help students learn numerals, number words, and beginning number concepts. Figure 2.19 is a screen display from the program that requires students to count the number of objects displayed and press the number on the keyboard. If the answer is correct, the program will play the familiar children's song, "Old MacDonald."



```
PLAYER 1        (MONEY EARNED: $0)
USE I, J, K & M TO MOVE; P TO PASS.
PRESS <SPACEBAR> TO ENTER MOPPET.
TYPE S TO STOP IF NO POSSIBLE MOVES.
```

FIGURE 2.18

The *Moptown Hotel* software program by The Learning Company.

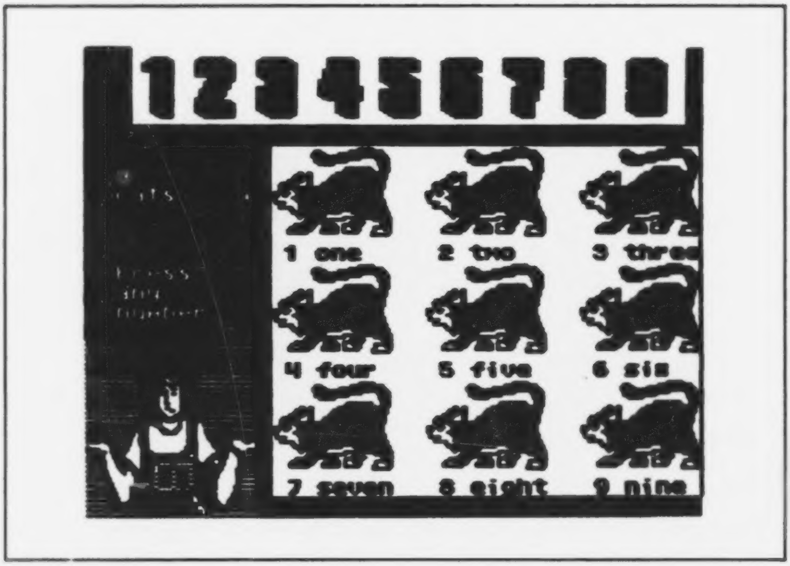


FIGURE 2.19

© DLM Teaching Resources, Allen, TX 75002.

Criticisms of Instructional Games

Computer games have been criticized because some teachers have been known to use them as babysitters, but it is also true that many computer games have violent or competitive themes that are unacceptable to some educators. These educators believe that the pseudo-destruction of people and planets in computer games may have a negative influence on students.

Computer games have also been criticized because they frequently require a learner to be adept at quickly locating and pressing keys on a computer keyboard to answer a question correctly. Rapid pressing of keys may be an extremely difficult task for a young or physically handicapped student. Such games requiring a high response rate from the user may actually be testing and improving motor skills rather than providing instruction.

Potential Value of Instructional Games

Instructional games are particularly well-suited to meeting objectives that involve recognition, discrimination, or repetitive drill. Game programs have the potential to motivate students to practice important skills that they might otherwise regard as tedious or boring.

A frequently cited benefit of video games is that they teach spatial visualization and eye-hand coordination. Other positive attributes of instructional games include (a) enabling a player to experience success while mastering a new situation and (b) using computer games to illustrate important computer processes.

Research on Effectiveness

Instructional games have been and will continue to be a popular approach to learning with a microcomputer; however, very little research has been done on their instructional effectiveness. Researchers have been more interested in investigating video games, and their preliminary findings indicate that video games may help to stimulate the chronically mentally ill and motivate reluctant learners.

In *Computers in Teaching Matehmatics,* (Kelman et al., 1983), Dr. Thomas Malone identified three characteristics of computer games that make them highly motivating for students: challenge, fantasy, and curiosity. However, there are no clear theories about what aspects of games promote learning or how they affect students. Until such questions are carefully researched, judgments about the instructional effectiveness of computer games are likely to be nothing more than ambiguous statements or testimonials by enthusiastic teachers.

CONCLUSION

The purpose of this chapter has been to provide useful information about the various categories of CAI. This information is essential because selecting and implementing educational software requires more than examining a program or completing an evaluation form. Selecting software for classroom use is similar to forming a business partnership. When you decide to form a partnership, you should know something about your partner's professional history, strengths, and weaknesses. Selecting software for instructional purposes creates a partnership among the student, the program, and the classroom teacher. Educators are obligated to understand the instructional characteristics of the materials they select—or suffer the consequences of a poor partnership.

Educators who use CAI programs should also realize that the field is still in its infancy. There are many unresolved questions about the instructional potential of CAI for microcomputers. Arielle Emmett (1983, p. 105) observed that:

> Thoughtful computer usage and understanding of what the blissful electronics can or can't do depends on work, on research, and on more sensitivity to a child's needs than perhaps we have, until now, comprehended. Understanding what makes a child learn better, and how, is crucial to any meaningful use of these machines. We simply don't know enough. And while remedying this is by no means an overnight task, it is certainly not the haphazard task that many have taken it to be. Computers are not simply Band-aids. Used well, they may be a significant component in solving our educational problems. But they are not the only component.

READINGS AND RESOURCES

Chambers, Jack A., and Sprecher, Jerry W. *Computer-Assisted Instruction.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

This is a comprehensive guide to developing CAI programs and to using microcomputers for providing course content instruction. It analyzes the effectiveness of CAI as a learning tool and specifically describes methods for using CAI in the classroom.

Coburn, Peter. *Practical Guide to Computers in Education.* Reading, MA: Addison-Wesley, Inc., 1982.

This comprehensive overview of CAI is one of several outstanding books written for the Addison-Wesley computer education series.

Greenblat, Cathy, and Duke, Richard. *Principles and Practices of Gaming-Simulation*. Beverly Hills, CA: Sage Publications, 1981.

This book presents a detailed overview of games and simulations. While it does not specifically focus on computer applications, much of the information presented can be generalized to microcomputer games and simulations.

Harper, Dennis, and Stewart, James. *RUN: Computer Education*. Monterey, CA: Brooks/Cole Publishing Co., 1983.

This collection of outstanding articles, written by experts in computer education, provides an easy-to-read overview of CAI as well as other aspects of educational computing.

Heinich, Robert; Molenda, Michael; and Russell, James D. *Instructional Media and the Technologies of Instruction*. New York: John Wiley & Sons, Inc., 1982.

This book surveys the entire field of instructional materials including CAI.

Hofmeister, Alan. *Microcomputer Applications in the Classroom*. New York: Holt, Rinehart, and Winston, 1983.

This is an outstanding resource for information about CAI. It stresses instructional considerations pertinent to the use of computers in education.

The following products have been included as examples of educational software in this chapter.

1. *Alien Addition* (Apple II+/IIe, TI 99/4A, IBM PC, Atari 800, Commodore 64)
   Animal Farm (Apple IIe)
   Publisher: Developmental Learning Materials
   One DLM Park
   Allen, TX 75002
2. *Music Theory* (Apple II+, IIe, Commodore 64, TRS Model III or 4, Atari 48K)

Discovery Lab (Apple II+, IIe, IIc)
Oregon (Apple II+, IIe, Commodore 64, TRS Model III or 4, Atari 48K)
Writing a Character Sketch (Apple II+, IIe)
Publisher: Minnesota Educational Computing Consortium
3490 Lexington Avenue North
St. Paul, MN 55112

3. *Know Your Apple IIe* (Apple IIe)
Publisher: MUSE Software
347 North Charles Street
Baltimore, MD 21201

4. *Microtype* (Apple IIe, IBM PC, TRS-80)
Publisher: South Western Publishing Company
5101 Madison Road
Cincinnati, OH 45227

5. *Mastertype* (Apple II+/IIe/III, Atari 400/800/1200, Commodore 64, IBM PC)
Publisher: Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591

6. *Rocky's Boots* (Apple II+/IIe)
Publisher: The Learning Company
545 Middlefield Road
Menlo Park, CA 94025

REFERENCES

Ahl, D. H. *101 BASIC Computer Games*. Maynard, MA: Digital Equipment Corporation, 1973.

Becker, H. J. *Microcomputers in the Classroom: Dreams and Realities*. Eugene, OR: International Council for Computers in Education, 1982.

Dennis, J. "Computer Simulation and Its Instructional Uses." *The Illinois Series on Educational Applications of Computers* 8E (1979):1–28.

Emmett, Arielle. "American Education: The Dead End of the 80's." *Personal Computing* 7 (1983):96–105.

Forman, D. "Search of the Literature." *The Computing Teacher* 9 (1982):37–49.

Garson, J. W. "The Case Against Multiple Choice." *RUN: Computer Education*, edited by D. O. Harper and J. H. Stewart. Monterey, CA: Brooks/Cole Publishing Company, 1983.

Greenblat, C., and Duke, R. *Principles and Practices of Gaming-Simulation*. Beverly Hills, CA: Sage Publications, 1981.

Kearsley, G.; Hunter, B.; and Seidel, R. "Two Decades of Computer Based Instruction Projects: What Have We Learned?" *T.H.E. Journal* 10 (1983):88–96.

Kelman, Peter; Bardige, Art; Choate, Jonathan; Hanify, George; Richards, John; Roberts, Nancy; Walters, Joseph; and Tornrose, Mary Kay. *Computers in Teaching Mathematics*. Reading, MA: Addison-Wesley, 1983.

Stallings, A., and Kaskowitz, D. *Follow Through Classroom Observation Evaluation*. Menlo Park, CA: Stanford Research Institute, 1974.

PROJECTS AND QUESTIONS

1. Define each term listed below.
    a. mainframe computer
    b. minicomputer
    c. microcomputer
    d. K
    e. RAM
    f. monitor
    g. dot matrix printer
    h. letter quality printer
    i. peripherals
    j. microchip
    k. floppy disk
    l. disk drive

2. List four categories of CAI programs. What are some advantages and disadvantages of each?
3. What is the difference between a tutorial program with a linear instructional design and a tutorial program with a branching instructional design? Draw and label diagrams that illustrate each design.
4. List some examples of instructional topics that do not readily conform to rules and procedures. Select one topic and write a short tutorial script for it. Specify exactly what will appear on the screen, the possible responses a student might make, and how the program would respond to correct and incorrect answers.

5. Interview a student or a teacher who has used computers for instruction. Report how the computer was used and include the user's perceptions about its strengths and limitations.
6. Locate and examine four different CAI programs. Describe your reaction to each program from an instructional perspective.
7. Read and summarize three articles that discuss research related to computers in education.
8. React to the following statements in writing or in a class discussion:
    a. The use of a computer to present drill-and-practice or tutorial programs is a waste of valuable resources.
    b. Computer learning limits social interaction among students.
    c. CAI programs will probably end up sitting on shelves like filmstrips and other audiovisual materials.
    d. Most educational software games are boring because students are used to playing more exciting games at home or at the video arcade.

3

Tool Software

INTRODUCTION

Educational software can be divided into three distinct categories. The first category consists of programs that use the computer to *improve* a student's performance in a particular skill or content area. The second category, simulations and games, uses the computer to *model* an environment; programs in this category encourage problem solving, social interaction, and many other kinds of learning. Programs for computer assisted instruction (CAI) comprise most of the software in these two categories.

A third category of educational software enables students and teachers to use the computer as a *tool*. Word-processing programs, record-keeping programs, and computer programming languages are a few examples of programs that allow the computer to be used as a tool. This chapter describes various tool programs used in education, presents examples of each, and discusses program features that should be considered when examining a program.

CLASSIFYING TOOL PROGRAMS

Tool programs can be divided according to their purpose. In general, tool programs are classified as follows:

- *General-Purpose Tools* These programs assist users by carrying out various tasks with content that is not specified ahead of time. The user determines the content, determines what needs to be done with it, and adapts the tool program to carry out the task. Examples of programs in this category include word-processing programs, database managers, and graphics and music utilities.
- *Special-Purpose Tools* These programs are designed to carry out tasks that are frequently repeated, for example, puzzle-making programs, readability programs, and programs that record and print students' grades.
- *Tool-Making Tools* These programs are broad in scope and have no immediate use other than to create other tools. All computer languages are examples of programs that serve this function.

GENERAL-PURPOSE TOOLS

Word Processing

Word-processing software allows a writer to create and manipulate text in ways that would be impossible with a typewriter. Imagine being able to insert text in the middle of a sentence or rearrange a series of paragraphs rapidly. Word-processing software enables a writer to do this and more.

A Brief History of Word Processing

Word-processing software specifically written for student use is a recent development. Word processing in the classroom is an out-growth of developments initially intended for business applications. It began in 1867 with the invention of the mechanical typewriter; however, the only thing that a modern word processor and a typewriter really have in common is a keyboard.

In 1964, IBM introduced the first Magnetic Tape Selectric Typewriter, a device that used a tape cartridge to store text. This typewriter was superior to an ordinary typewriter because it allowed a writer to record, erase, and rapidly search for information. Electronic typewriters linked to a video display emerged in the early 1970s and offered much faster editing than typewriters.

The first floppy disks for use with word processors appeared in 1973. Floppy disks replaced the tape cartridge as a storage system, allowing vast amounts of text to be saved or retrieved almost instantly.

Microprocessors, complete "electronic brains" on a computer chip, were incorporated into word-processing products in the mid-1970s. Unlike today's microcomputers, these early word-processing computers were dedicated to word processing only and could perform no other tasks; however, such "dedicated" word processors are still widely used in business.

Today there are word-processing programs for almost every brand of microcomputer, but because most word-processing software is not written with students in mind, these programs have characteristics that make them difficult to master. In addition, they include features that are not necessary for most students' writing needs.

Features to Consider

All word processors perform certain basic functions; however, not every word processor performs them in the same way. For this reason it is important to select a word-processing program that will enable students to use the program with ease. The functions described below determine how quickly and easily a student can write or edit a document with a word processor.

Text Editing Functions

- *Writing and editing in one mode.* Programs that allow the writer to compose and edit without changing screens or modes are easier for students to use.
- *Replacing text.* There are several ways to replace text. The "overwrite" method allows the writer to replace words or letters by typing over the existing text. The "insert/delete" method requires that the writer delete old text and insert new text. The advantage of the insert/delete method is that the writer can make a revision on the screen, compare it to the old text, and then choose which text should be deleted.
- *Search and replace.* Some programs have a feature that allows the writer to search for a word or phrase and replace the word/phase wherever it appears in a document. This is particularly useful for helping students locate and revise misspelled or redundant words. It is also helpful for teachers who use a word processor to produce forms, texts, or reports that may be used for different purposes but require only minor changes.
- *Deleting text.* Powerful word-processing programs permit the writer to delete text by character, word, sentence, line, paragraph, or entire file.
- *Moving blocks of text.* Most word-processing programs can move a sentence, paragraph, or large block of text to a different location in a document. Programs that allow the writer to indicate the beginning and end of text to be move as well as the new location are the easiest to use.
- *Moving the cursor.* A word-processing program should allow the writer to move the cursor up, down, and across the screen with ease.

Screen Display and Formatting Functions

- *Screen display and printed text.* Screen displays are usually described by the number of characters that can be displayed on the screen at one time. Many word-processing programs display 24 lines and up to 80 columns on the screen at one time. Such programs are sometimes called "80-column" programs. Other programs may display fewer columns and hence fewer characters on the screen. Because the size of letters in an 80-column program is smaller than in a 40-column program, 40-column program displays may be easier for young children to read.

Some word-processing programs do not print text exactly as it appears on the screen. This may cause a problem for a writer who wants to print columns of information.

Remember, a good quality monitor is necessary to display text created with a word-processing program clearly. Using an 80-column word-processing program with a conventional television set or a color monitor can result in text that is difficult to read; therefore, a monochrome monitor is recommended.

- *Text formatting.* Sophisticated word-processing programs allow writers to type continuously without having to lift their fingers from the keyboard to press a carriage-return key. When the cursor reaches the right edge of the screen or the end of a line, it automatically moves to the beginning of the next line and moves a word with it if there is not enough room for it at the end of the line. This feature is referred to as automatic "word wrap." Powerful word-processing programs will also close up spaces and adjust line widths when a writer inserts or deletes text.
- *Scrolling.* Scrolling allows the text on the screen to move up or down, usually by moving the cursor up or down to reveal text above or below a certain point. A useful feature is the ability to move to the beginning and end of a document immediately without scrolling.

Print Formatting Functions

Print formatting refers to the part of a word-processing program that directs a printer to arrange text on paper. Formatting features include options for double- or single-spacing, adjusting margins, underlining words, centering text, and numbering pages.

Some word-processing programs ask the user to respond to formatting questions specified on a menu. More sophisticated programs allow the user to direct the printer by specifying commands that are embedded within a document. For example, .LM 10 followed by a line of text is an embedded command that directs a printer to set the left margin to 10 when the *Apple Writer IIe* program is used. Embedded commands allow the writer to make print changes without leaving the typing area; but these kinds of commands are difficult for students to remember.

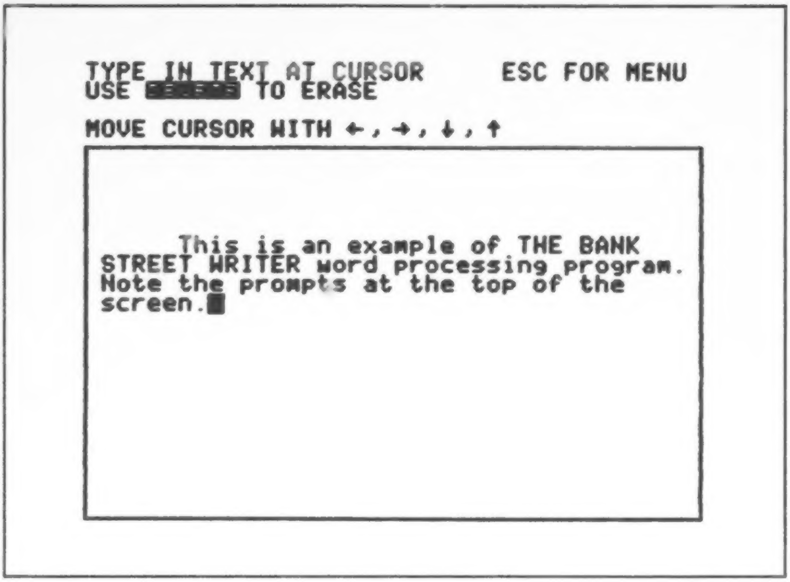
Other Important Features

- *Single keystroke commands.* Many word-processing programs require students to press two keys simultaneously to perform a function. Programs that use single keystroke commands are easier for elementary or physically handicapped students.
- *Help features.* Programs that provide "Help" options, prompts on the screen, and concise reference cards are easier for all students to use.
- *Safety features.* Highlighting important information and requiring the user to verify whether or not a function should be completed can prevent the accidental removal of text. Programs that prompt the writer with questions such as, "DO YOU REALLY WANT TO DELETE THIS TEXT Y/N," require a writer to pause and think before making an important decision.

Examples of Word-Processing Programs

Figure 3.1 shows a screen display from *The Bank Street Writer*, a popular word-processing program designed for student use. This program gives prompts to assist the user with specific word-processing functions, which virtually eliminates the need for the printed manual.

Figure 3.2 shows a screen display from the *Apple Writer IIe* word-processing program, a sophisticated program suitable for use by high school students, college students, and their teachers. The program's many features include a word-processing language



```
TYPE IN TEXT AT CURSOR        ESC FOR MENU
USE DELETE TO ERASE

MOVE CURSOR WITH ←, →, ↓, ↑

     This is an example of THE BANK
STREET WRITER word processing program.
Note the prompts at the top of the
screen.█
```

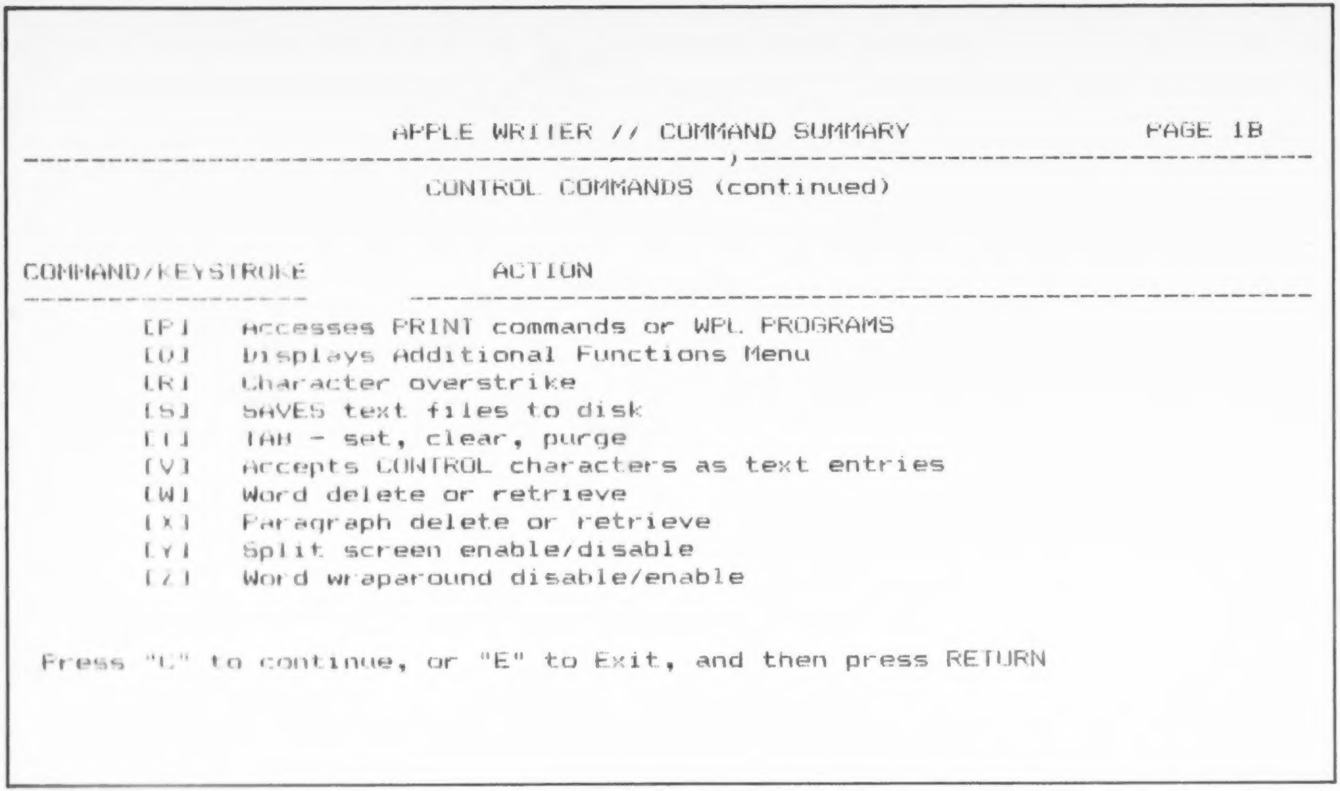
FIGURE 3.1

© Scholastic Inc. All rights reserved. Used by permission of Scholastic, Inc.

for creating customized documents, a glossary that allows a writer to store and recall repetitive phrases, and a split screen display for comparing text in different parts of a document.

Creative Writing

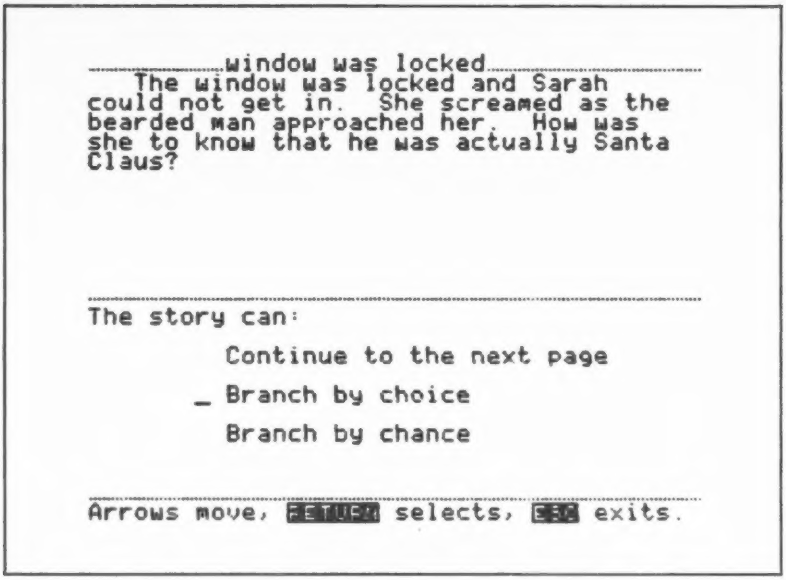
There are also programs that assist students with creative writing by asking a series of interactive questions that prompt them as they write. *Storytree* is a program for reading and writing interactive stories or a story that unfolds according to choices made by the writer. Stories written with *Storytree* branch from one beginning to many different possible endings. Figure 3.3 shows a screen display from *Storytree*.



```
                         APPLE WRITER // COMMAND SUMMARY                    PAGE 18
----------------------------------------------)-------------------------------------
                              CONTROL COMMANDS (continued)


COMMAND/KEYSTROKE                 ACTION
-----------------           ----------------------------------------------------
      [P]    Accesses PRINT commands or WPL PROGRAMS
      [Q]    Displays Additional Functions Menu
      [R]    Character overstrike
      [S]    SAVES text files to disk
      [T]    TAB - set, clear, purge
      [V]    Accepts CONTROL characters as text entries
      [W]    Word delete or retrieve
      [X]    Paragraph delete or retrieve
      [Y]    Split screen enable/disable
      [Z]    Word wraparound disable/enable


 Press "C" to continue, or "E" to Exit, and then press RETURN
```

FIGURE 3.2

Reproduced with the permission of Apple Computer, Inc.



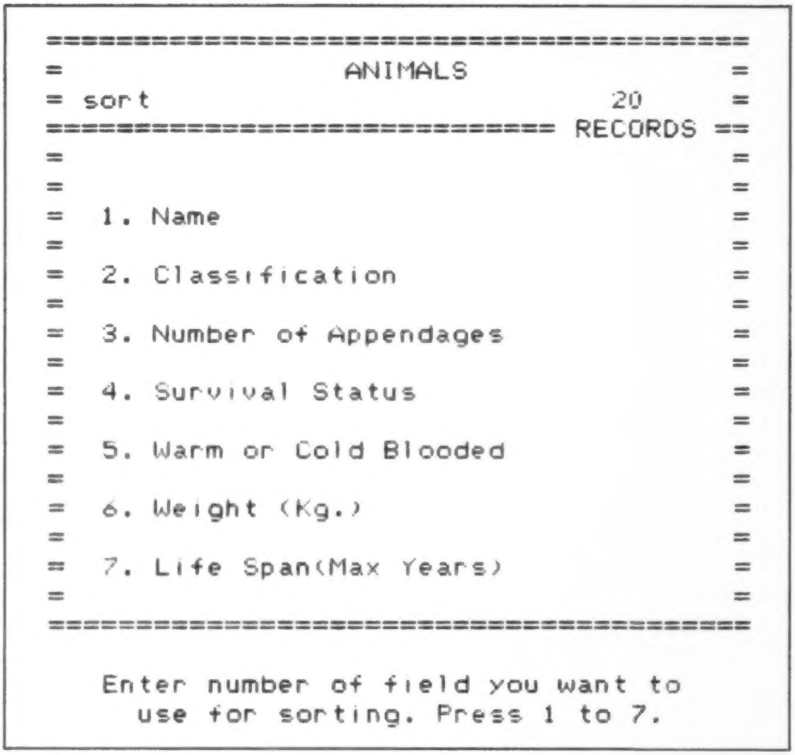
```
............window was locked..................
  The window was locked and Sarah
could not get in.  She screamed as the
bearded man approached her.  How was
she to know that he was actually Santa
Claus?

...............................................
The story can:

            Continue to the next page

         _ Branch by choice

            Branch by chance

...............................................
Arrows move, RETURN selects, ESC exits.
```

FIGURE 3.3

© Scholastic Inc. All rights reserved. Used by permission of Scholastic, Inc.

Information Processing Tools

Collections or groups of data are called a database. For example, all the names, addresses, and phone numbers of students in a school would form a database. A special program, called a database program, can be used to create, organize, alter, and print information that constitutes a database. Students might use such a program to organize a list of their favorite rock groups or items in their stamp or baseball card collections. For students growing up in an information-rich age, using a database program may provide them with an understanding of how information can be organized, stored, and retrieved. Figure 3.4 shows a screen display from *Friendly Filer,* a database program suitable for use with elementary and middle school students.



```
========================================
=                ANIMALS               =
= sort                        20       =
============================ RECORDS ==
=                                      =
=                                      =
=  1. Name                             =
=                                      =
=  2. Classification                   =
=                                      =
=  3. Number of Appendages             =
=                                      =
=  4. Survival Status                  =
=                                      =
=  5. Warm or Cold Blooded             =
=                                      =
=  6. Weight (Kg.)                     =
=                                      =
=  7. Life Span(Max Years)             =
=                                      =
========================================

   Enter number of field you want to
     use for sorting. Press 1 to 7.
```

FIGURE 3.4

Screen display from *Friendly Flyer* by Grolier Electronic Publishing.
Copyright 1984 by Intentional Educations, Inc.

Before purchasing a database program, you must decide what kind of information you want to track, how much money you can afford to spend, and how much time you're willing to devote to learning a program. Some important considerations for selecting a database program include:

- *Is it easy to design a form with the program?* To use a database filing program, you will need to create a "data entry" form containing keywords, which you will define. Each data entry form is called a record. "Phone Number" is an example of a keyword that you might use in a database program to record the phone numbers of students in a class. You will want to select a program that allows you and your students to create, copy, and modify a form easily.

- *How many keywords can you include in a record?* A record consists of a single file with a list of keywords. Some programs limit the number of keywords that can be listed in a record. The more details you or your students want to record, the more keywords you will need.
- *How does the program print and display records?* You should find out if the program requires you to print an entire record or part of a record. Some programs provide the option to display individual records on the screen either one at a time or as a group.
- *How does the program search for data?* Can you locate a record from a group of records by instructing the program to search for a particular piece of information?
- *How does the program sort records?* Does the program sort from A to Z, Z to A, 0 to 9, or 9 to 0? Find out if the program allows you to sort any keyword listed on a form. You should also check how long it takes the program to perform a sort.

Another useful software tool for processing information is an electronic spreadsheet, which allows the easy manipulation of numerical data. A spreadsheet program uses a monitor to display a large, imaginary sheet of paper that is divided into columns and rows like standard ledger paper; however, a spreadsheet is more like "superpaper" because it allows numbers to be manipulated quite easily. Spreadsheet programs are often used to solve problems in business, science, and engineering.

There are a number of database and spreadsheet programs on the market. Most are designed with adults in mind and are not easy for students to use. An important point to remember when evaluating any software, however, is that learning ease is relative to a user's experience level. The more experience an evaluator has with specific types of software—and software in general—the less time it takes for that individual to learn how to use a program. This accounts for different evaluations of a program by different evaluators. Rating is frequently influenced by perspective, experience, and an individual's ability to persist in a new situation.

Tool Programs for the Creative Arts

Songwriter is an example of a program that enables a microcomputer to be used for exploring musical concepts while composing songs. Figure 3.5 illustrates *Songwriter*'s composing screen, a keyboard with the commands needed to compose a melody and prompts to remind the student what the commands are. Arrow keys are used to move the cursor on the keyboard, and notes are "recorded" by pressing the space bar. When a melody is played, the screen display resembles a paper piano roll for a player piano. The program also allows modules of musical ideas to be created and repeated as well as note names to be printed in songs.

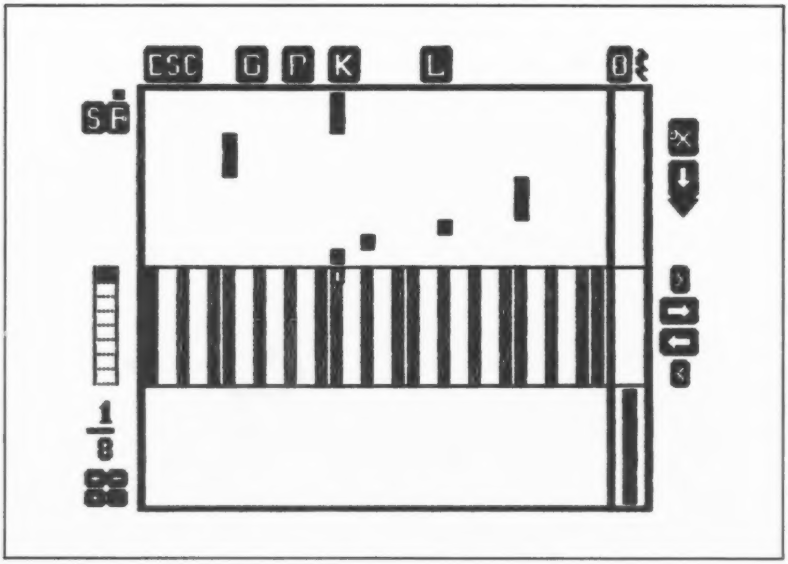


FIGURE 3.5

© 1983–1985 Scarborough Systems, Inc., Tarrytown, NY.

Songwriter's design resembles *The Bank Street Writer* word-processing program. It is a music processor that provides prompts and interactive messages, virtually eliminating the need for the user to refer to a manual to learn how to use the program.

The Koala Pad is a pressure-sensitive tablet used with special software to produce graphics. It converts finger or stylus pressure into electronic signals for controlling a computer, and is particularly useful for drawing pictures and pointing to images on computer monitors. A finger movement across the tablet could result in the drawing of a colored line, the movement of a game piece, or the creation of a sound.

Figure 3.6 illustrates a screen display from the *Koala Micro Illustrator*, a program specifically for creating computer graphics with the Koala Pad. The user selects options from the menu by touching specific areas of the Koala Pad, then presses a button on the pad to clear the screen for drawing.

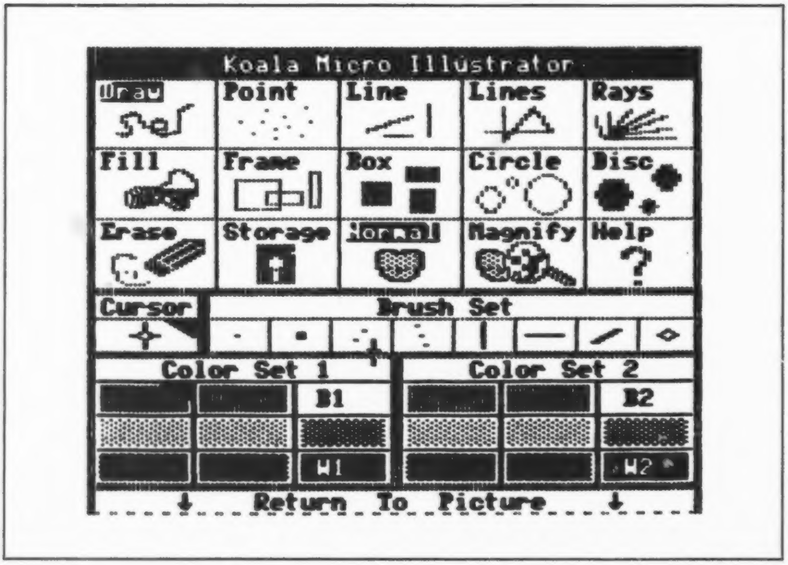

Koala Micro Illustrator
Draw
Point
Line
Lines
Rays
Fill
Frame
Box
Circle
Disc
Erase
Storage
Normal
Magnify
Help
Cursor
Brush Set
Color Set 1
Color Set 2
B1
B2
W1
W2
Return To Picture

FIGURE 3.6

Touch tablets like the Koala Pad do not require students to be adept at typing. As an alternative to interacting with a computer via the keyboard, touch tablets are well-suited for use with primary-grade students and students with physical handicaps.

The *Micro Illustrator* software includes icons, pictorial representations of a word or idea. For example, some word-processing programs display a file cabinet to represent the word "file." To activate the FILE command, the user would simply move the cursor over the icon or picture of the file cabinet. Icons are helpful to students with poor reading skills and are being incorporated into a number of sophisticated programs as a means of making software readily understandable to the novice user.

Figure 3.7 shows a screen display from *MacWrite,* a word processing program for the Apple MacIntosh computer. A small device called a mouse, which is connected to the computer by a cable, is used to interact with the program. A cursor moves around the screen when the mouse is moved. Menus, or choices for action, appear on the screen as icons, and the user makes a selection by moving the cursor over the icon and clicking a button on the mouse.

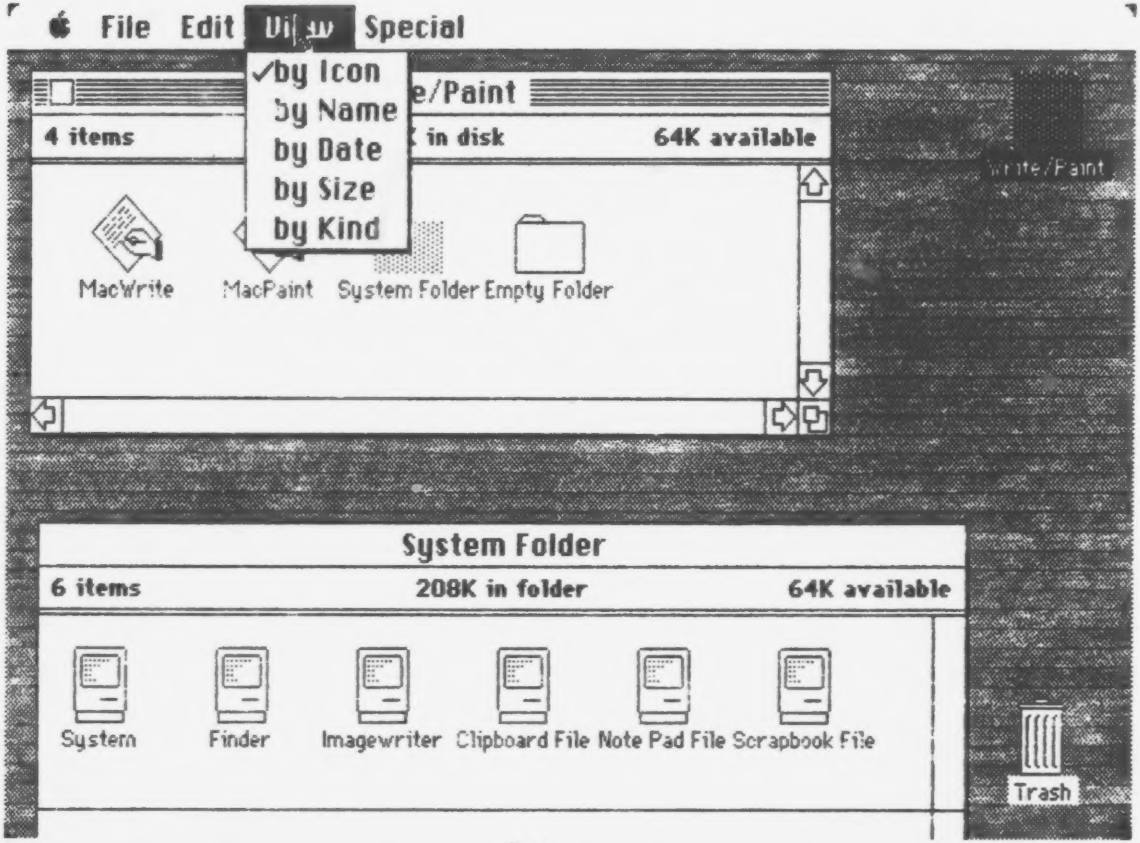
Integrated Tool Programs

Programs referred to as integrated software combine the functions of several tool programs into a single package. *Appleworks* is an example of an integrated program that allows the user to select from word-processing, database, or spreadsheet programs (see Figure 3.8). An integrated program can reduce the cost of purchasing three separate tool programs and eliminate the need to memorize different commands for each program.

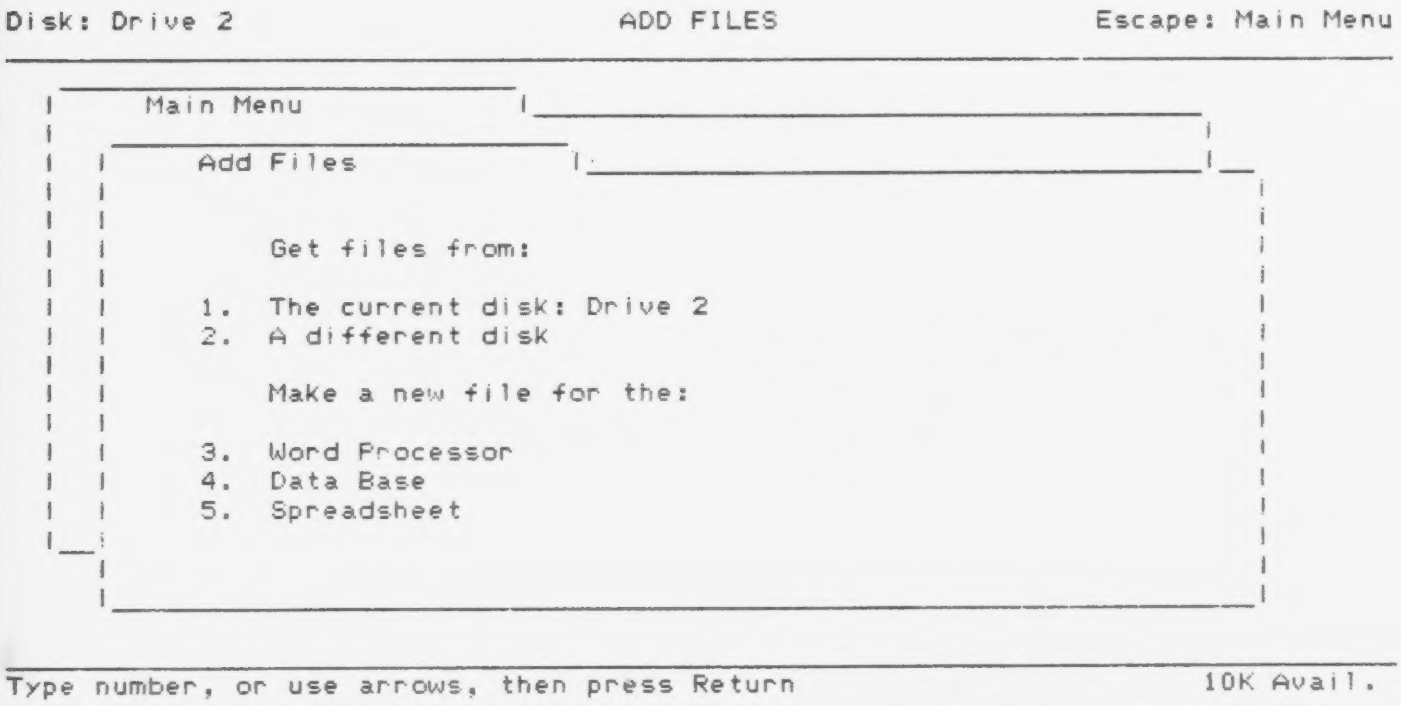
SPECIAL-PURPOSE TOOLS

Special-purpose programs are used by educators to create instructional materials or to process specific data. The following section presents examples of special-purpose programs used by teachers and students.

FIGURE 3.7



File
Edit
View
Special
✓by Icon
by Name
by Date
by Size
by Kind
e/Paint
4 items
K in disk
64K available
MacWrite
MacPaint
System Folder
Empty Folder
Write/Paint
System Folder
6 items
208K in folder
64K available
System
Finder
Imagewriter
Clipboard File
Note Pad File
Scrapbook File
Trash



```
Disk: Drive 2                        ADD FILES                      Escape: Main Menu
_____________________________________________________________________________________

  |    Main Menu          |________________________________________
  |                                                                |
  |  |     Add Files         |_________________________________    |__
  |  |                                                               |
  |  |                                                               |
  |  |         Get files from:                                       |
  |  |                                                               |
  |  |     1.  The current disk: Drive 2                             |
  |  |     2.  A different disk                                      |
  |  |                                                               |
  |  |         Make a new file for the:                              |
  |  |                                                               |
  |  |     3.  Word Processor                                        |
  |  |     4.  Data Base                                             |
  |  |     5.  Spreadsheet                                           |
  |__|                                                               |
     |                                                               |
     |_______________________________________________________________|

_____________________________________________________________________________________
Type number, or use arrows, then press Return                          10K Avail.
```

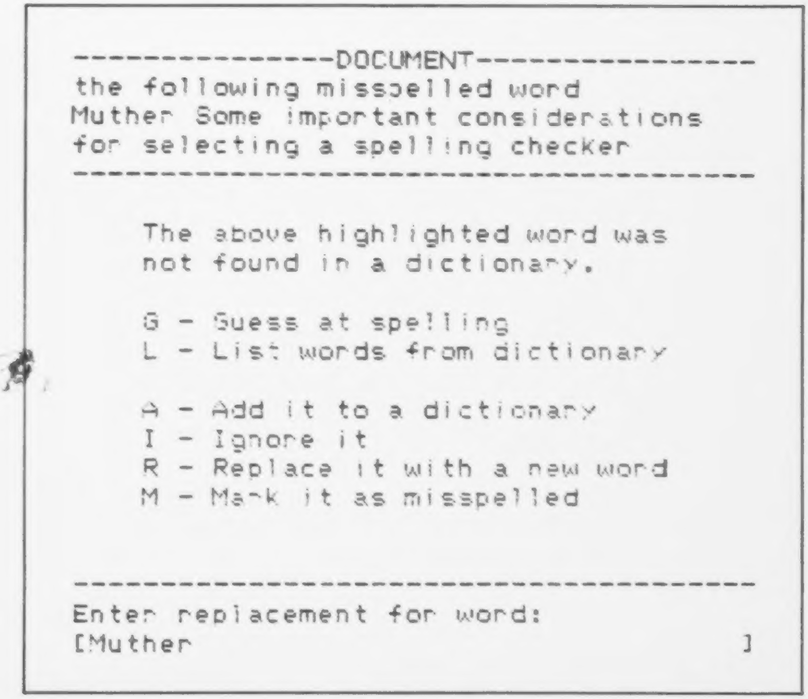
FIGURE 3.8

Reproduced with the permission of Apple Computer, Inc.

Spelling Checkers

Spelling checkers are programs that read every word in your text for comparison against an electronic dictionary. Words that do not match are considered suspect and are flagged for the user's consideration. Figure 3.9 shows a screen display from *Sensible Speller,* a spelling checker program. Notice how the program reports the misspelled word, Muther. Here are some important considerations for selecting a spelling checker program.

- *Can you integrate the spelling checker with your word-processing program?* Every spelling checker program cannot be used with every word-processing program. Before you purchase a



```
---------------DOCUMENT----------------
the following misspelled word
Muther Some important considerations
for selecting a spelling checker
---------------------------------------

     The above highlighted word was
     not found in a dictionary.

     G - Guess at spelling
     L - List words from dictionary

     A - Add it to a dictionary
     I - Ignore it
     R - Replace it with a new word
     M - Mark it as misspelled


---------------------------------------
Enter replacement for word:
[Muther                               ]
```

FIGURE 3.9

spelling checker, you should determine if the program will work with your word-processing software.

- *How fast is the program?* Some spelling checkers can search an entire document for suspected misspellings in less than thirty seconds, while other programs may take as long as two minutes. A spelling checker's speed is related to the number of words in a document. It is also important to find out if there are any limits on the number of words or pages a program can check at one time.
- *What is the size of the program's dictionary?* The heart of any spelling checker is its dictionary. A dictionary with 20,000 to 30,000 words usually is sufficient for most writers.
- *Can you modify the program's dictionary?* It is not likely that a spelling checker program will include all of the words that you or your students may misspell. You should try to select a program that allows you to add your own word lists. If a program can be modified, you should also find out how many words can be added.
- *How does the program identify a misspelled word?* Some spelling checkers will mark a suspected misspelling in the text, while other programs will present a list of suspected words.
- *Does the program or the user correct the misspelled words?* Some programs automatically change a misspelled word wherever it occurs in the text. If you purchase such a program, make sure that the program is an accurate editor! Other programs present the correct spelling of a word, but the user must correct the misspelling.

Graphing Programs

Programs that allow the user to input data and produce bar graphs can be useful for teachers and students. Such programs can rapidly process large amounts of data and clearly print or display the results. A graphing program might be used in the classroom to summarize student test scores or data from math, science, or social studies projects. Figure 3.10 shows a screen display from *PFS:Graph*, a program that allows the user to create bar, pie, or line graphs.

Readability Analysis Programs

A readability analysis program can be used by educators to obtain information about the approximate reading level of printed materials. Researchers have developed readability formulas that predict the difficulty of material for readers at a given level. The level of difficulty is usually expressed in terms of a school grade level.

How can a computerized readability program assist the teacher? Computers make possible:

1. Rapid, accurate, and simple assignment of grade levels to printed materials.
2. Printouts of unfamiliar words by a readability formula.

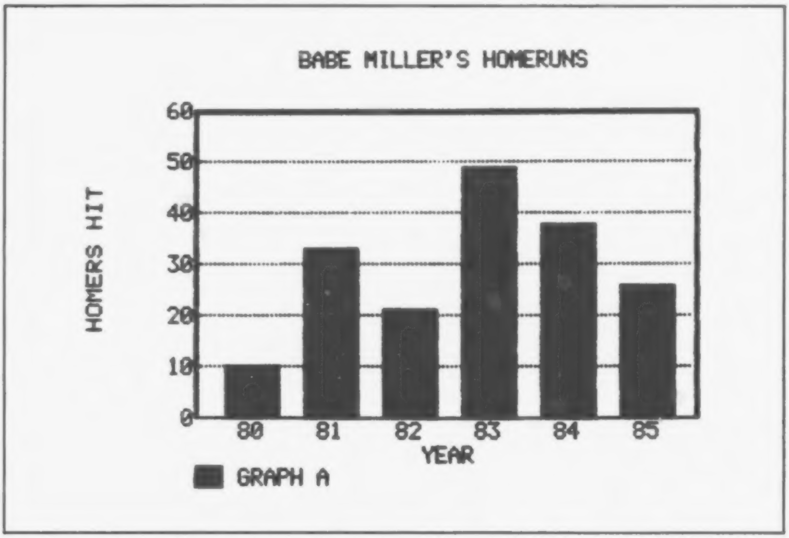
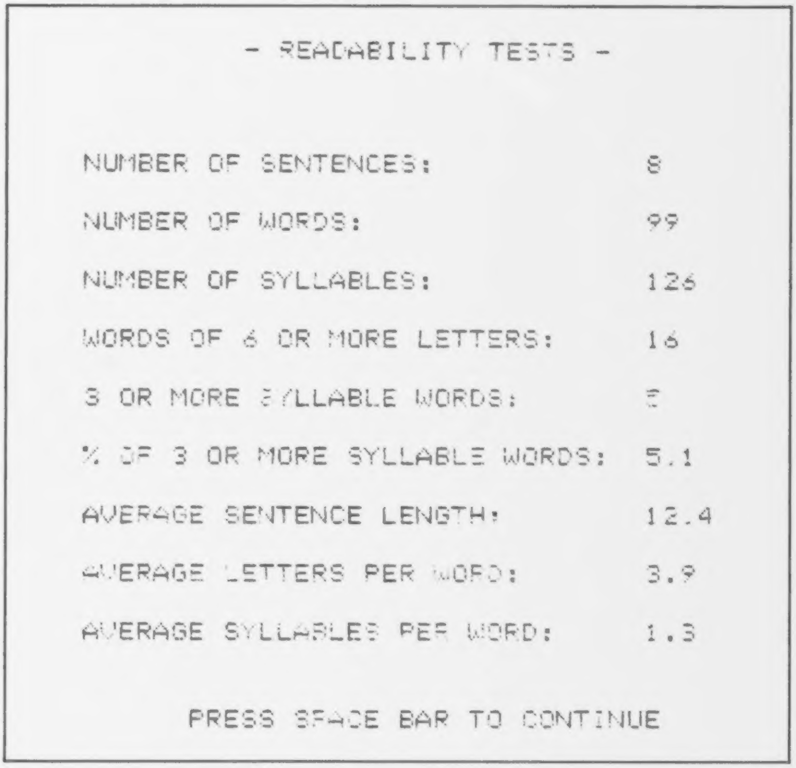


FIGURE 3.10

PFS®: GRAPH, courtesy of Software Publishing Corporation.

3. Prescriptive teaching by enabling educators to assign accurately labeled materials according to an individual student's needs.
4. The sequencing of basal reading stories in order of difficulty.

Figure 3.11 shows a screen display from Readability Tests. Note the amount of information the program presents about a specific reading passage. If you decide to use a readability program, remember that each readability formula is based on differ-



```
            - READABILITY TESTS -


NUMBER OF SENTENCES:                 8

NUMBER OF WORDS:                     99

NUMBER OF SYLLABLES:                 126

WORDS OF 6 OR MORE LETTERS:          16

3 OR MORE SYLLABLE WORDS:            5

% OF 3 OR MORE SYLLABLE WORDS:       5.1

AVERAGE SENTENCE LENGTH:             12.4

AVERAGE LETTERS PER WORD:            3.9

AVERAGE SYLLABLES PER WORD:          1.3

         PRESS SPACE BAR TO CONTINUE
```

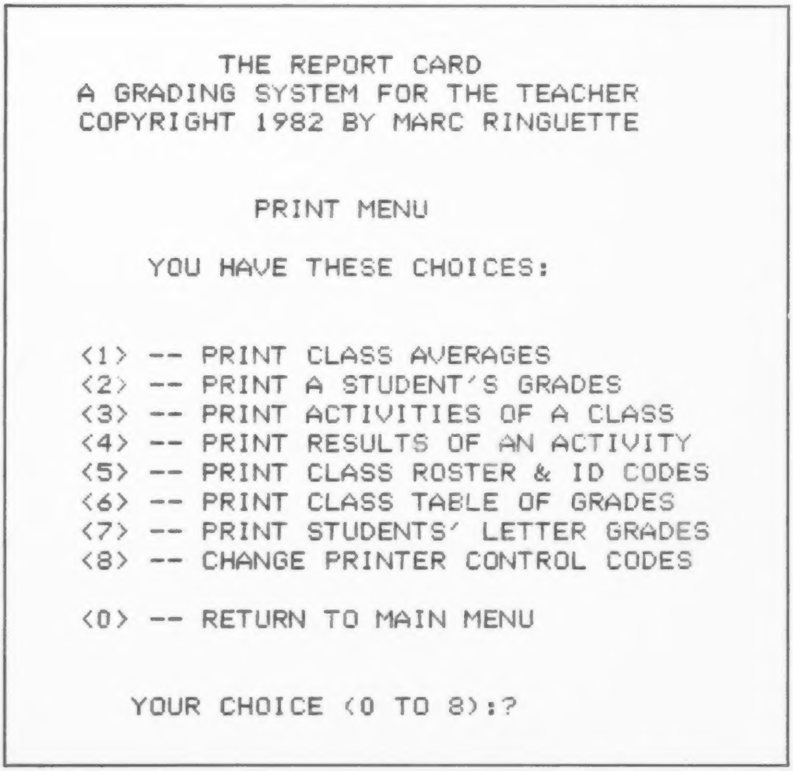
FIGURE 3.11

This has been taken from Readability Tests published by MECC,
3490 Lexington Avenue North, St. Paul, MN 55112.

ent language variables to provide an index of probable difficulty for readers. You should select a program that allows you to enter and edit a reading passage easily and select from a variety of readability formulas.

Gradebook Programs

A typical teacher may record and summarize thousands of students' grades or scores. A gradebook program allows grades or scores to be entered and quickly summarized. Figure 3.12 shows a screen display from *The Report Card,* a gradebook program that



```
            THE REPORT CARD
     A GRADING SYSTEM FOR THE TEACHER
     COPYRIGHT 1982 BY MARC RINGUETTE


               PRINT MENU

          YOU HAVE THESE CHOICES:


     <1> -- PRINT CLASS AVERAGES
     <2> -- PRINT A STUDENT'S GRADES
     <3> -- PRINT ACTIVITIES OF A CLASS
     <4> -- PRINT RESULTS OF AN ACTIVITY
     <5> -- PRINT CLASS ROSTER & ID CODES
     <6> -- PRINT CLASS TABLE OF GRADES
     <7> -- PRINT STUDENTS' LETTER GRADES
     <8> -- CHANGE PRINTER CONTROL CODES

     <0> -- RETURN TO MAIN MENU


         YOUR CHOICE (0 TO 8):?
```

FIGURE 3.12

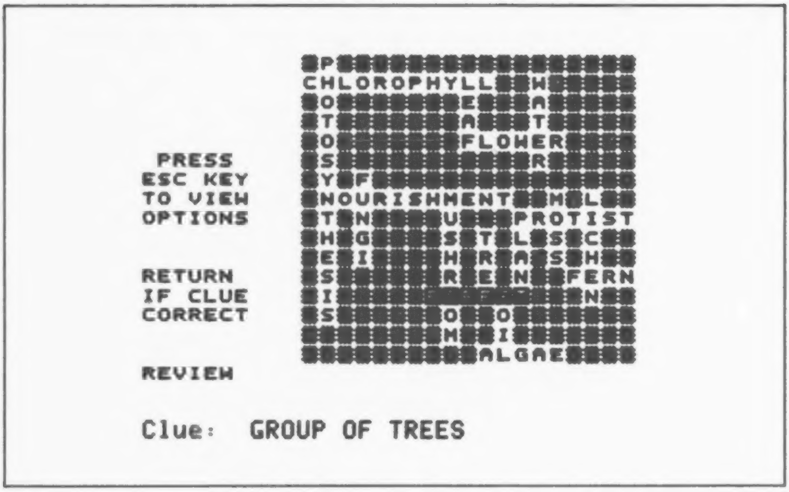
stores, sorts, summarizes, and prints grades and scores. Some important considerations for selecting a gradebook program include:

- *How many classes and students will the program support?* A gradebook program should allow you to enter all of your students and classes on a single disk.
- *Does the program allow data to be entered, deleted, or modified easily?* If you select a gradebook program that requires a lot of time to enter grades, you may as well use a conventional gradebook. A gradebook program should allow you to create a class list and enter or change grades quickly.
- *How does the program sort and report data?* A gradebook program should rapidly sort and print student data by grade in school, first name, last name, or student identification code. It should also perform statistical calculations such as total scores, number of completed assignments, and individual and class averages. The program should allow records to be either displayed on the screen or printed. Another useful feature is the option to print comments along with each grade report.

Puzzle- and Poster-Making Programs

Among the most popular software used by educators are programs that enable teachers and their students to create crossword puzzles, word searches, and posters. Such programs are often used to reward students and to provide alternative practice material in various subject areas.

Crossword Magic creates crossword puzzles on the screen and on paper using words and clues entered by a teacher or student (see Figure 3.13). *Word Find* allows the user to create word-search puzzles (see Figure 3.14). The program arranges words horizontally, vertically, or diagonally into an array of letters that hide the target words. Both *Crossword Magic* and *Word Find* allow the user to determine the size of a printed puzzle. They also supply a separate answer key.



PRESS
ESC KEY
TO VIEW
OPTIONS

RETURN
IF CLUE
CORRECT

REVIEW

Clue: GROUP OF TREES

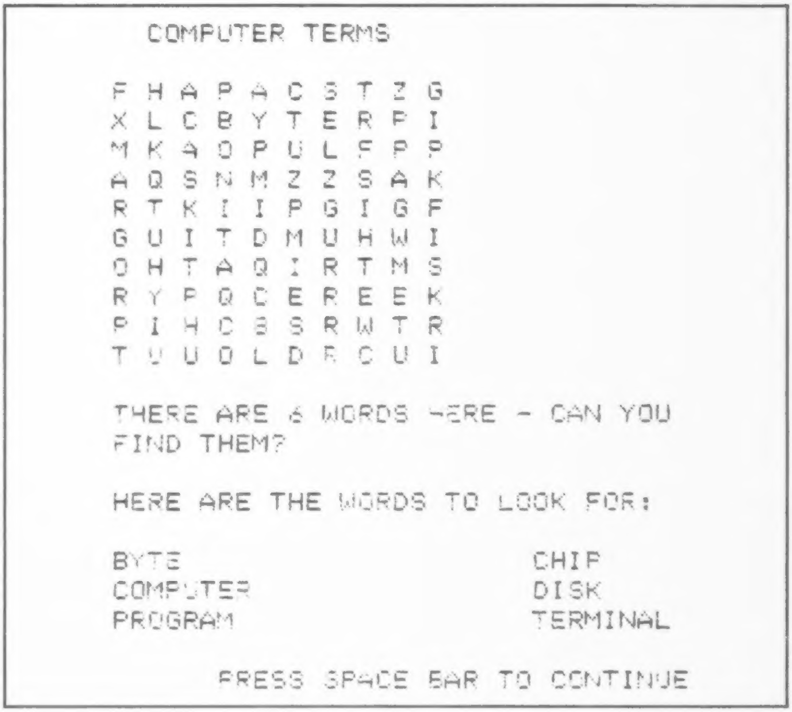
FIGURE 3.13

Teachers frequently spend hours creating posters for bulletin boards, signs, or positive reinforcement notes. *The Printshop* enables a teacher or student to view, select, and assemble various elements from the program's library of type styles and pictures in order to create and print personalized, decorative messages (see Figure 3.15).

TOOL-MAKING TOOLS

As computer assisted instruction has emerged, computer experts have begun to ask: What are the appropriate uses of computers in education? Should the computer teach the student or should the student teach the computer?

In drill-and-practice, simulation, tutorial, or game programs, students acquire knowledge from the computer. Tool-making programs (such as computer programming languages) enable students to create their own unique programs—in effect, they are programs that can be used by students to teach the computer.



```
   COMPUTER TERMS

F H A P A C S T Z G
X L C B Y T E R P I
M K A O P U L F P P
A Q S N M Z Z S A K
R T K I I P G I G F
G U I T D M U H W I
O H T A Q I R T M S
R Y P Q C E R E E K
P I H C B S R W T R
T U U O L D R C U I

THERE ARE 6 WORDS HERE - CAN YOU
FIND THEM?

HERE ARE THE WORDS TO LOOK FOR:

BYTE                         CHIP
COMPUTER                     DISK
PROGRAM                      TERMINAL

            PRESS SPACE BAR TO CONTINUE
```

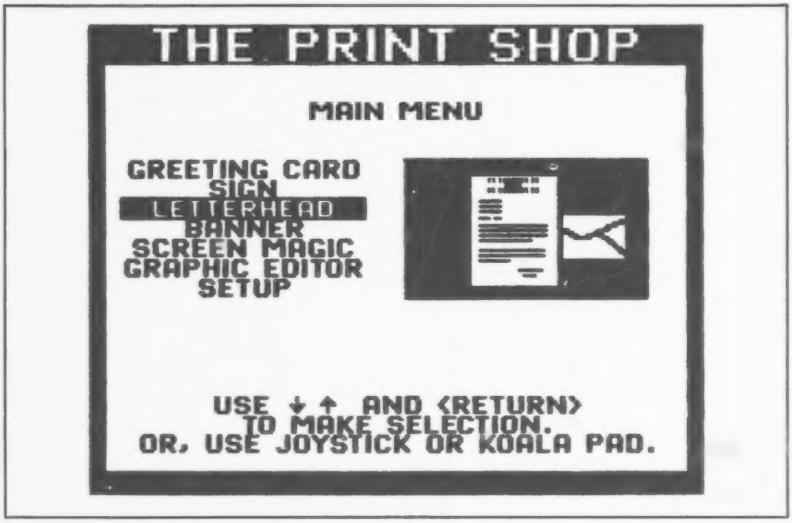
FIGURE 3.14

This has been taken from Word Find, published by MECC,
3490 Lexington Avenue North, St. Paul, MN 55112.

A computer program is the instructions that specify the sequence of steps the computer must follow to complete a specific task. A complete set of programming instructions is referred to as a programming language. There are a number of programming languages being used in the schools, but among the most popular are BASIC, LOGO, Pascal, and PILOT. The next section briefly describes each of these languages.

BASIC

BASIC stands for Beginners All-purpose Symbolic Instruction Code. It was developed as a teaching tool by John Kemeny and



USE ↓ ↑ AND <RETURN> TO MAKE SELECTION. OR, USE JOYSTICK OR KOALA PAD.

FIGURE 3.15

The Print Shop™ is a trademark of Broderbund Software, Inc. © 1984.

Thomas Kurtz at Dartmouth College in 1965, and it is frequently used to train students how to program. BASIC allows the programmer to write commands in order of their execution.

There are many versions of BASIC, each with different features and capabilities. Most microcomputers have the BASIC language "built in" to the computer. This means that the computer is ready to be programmed in BASIC when it is switched on. Programming languages other than BASIC are usually not machine-resident. A disk with the programming language on it must be purchased and then loaded into the computer—if that computer has the appropriate hardware requirements necessary to use the language.

BASIC is quite popular with individuals who program for recreation; however, it has several disadvantages. First, there is no commonly accepted form of BASIC that can be used with every brand of microcomputer. Second, BASIC programs can be difficult to modify or correct because the language does not require a programmer to program in a structured form. A third disadvantage of BASIC is its speed. BASIC is usually not processed as rapidly by the computer as other programming languages. A sample BASIC program appears below.

```
10  REM   SAMPLE PROGRAM
20  A = 0
30  PRINT  “ENTER 3 NUMBERS, PRESS RETURN AFTER EACH”
40  FOR I = 1 TO 3
50  INPUT N
60  IF N > A THEN A = N
70  NEXT I
80  PRINT  “THE LARGEST NUMBER WAS ” ; A
90  END
```

Pascal

Pascal (named for Blaise Pascal, the seventeenth-century French mathematician) was developed by Niklaus Wirth and became available in 1971. Pascal is taught in many computer science programs at universities. It allows the programmer to write complex programs as a series of simpler procedures.

A major advantage of Pascal is its “portability.” Pascal can be programmed or coded in a way that is relatively standard and, therefore, is more compatible for use with different computers. Pascal promotes programming in a structured format that is understandable by other programmers, and it is easy to modify; however, Pascal usually requires significantly more time to learn than BASIC. A sample Pascal program appears below.

```
PROGRAM LARGEST
VAR
  BIG, TRY, COUNT : INTEGER;
BEGIN
  BIG : = 0 ;
  WRITELN (‘ENTER 3 NUMBERS, PRESS RETURN AFTER EACH’);
  FOR COUNT : = 1 TO 3 DO
  BEGIN
    READLN (TRY);
    IF TRY > BIG THEN BIG : = TRY
  END
  WRITELN (‘THE LARGEST NUMBER WAS ,BIG);
END
```

PILOT

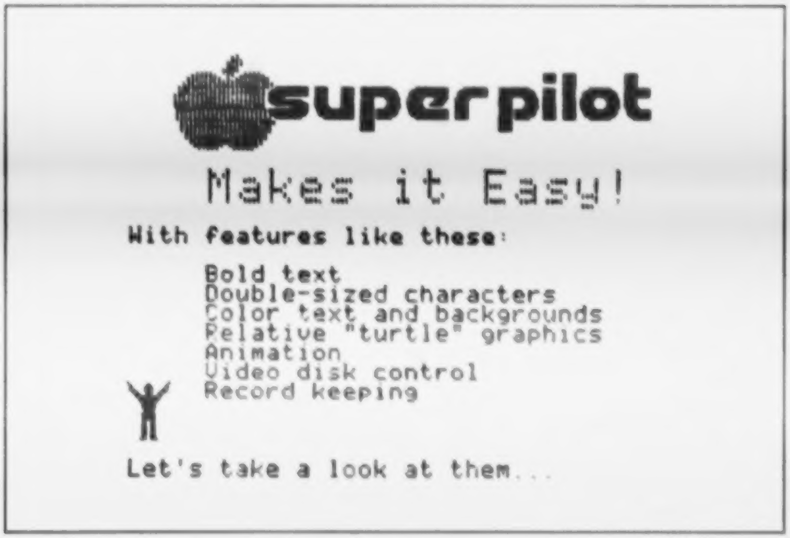
PILOT (Programmed Inquiry Learning Or Teaching) was originally developed at the University of California at San Francisco in the late 1960s to create dialogue-type material for use with college students. PILOT is classified as an authoring language because it is used to create CAI programs. Figure 3.16 shows a screen display from the *Superpilot* program. A disadvantage of PILOT is that it is specifically limited to programming CAI materials. Another disadvantage is that newer versions of the language are quite complex because features have been added that allow sound, color, and graphics to be used in programs; these features require teachers to learn additional programming commands that further complicate the task of creating an instructional program. A sample PILOT program appears below.

```
        R:  SAMPLE PILOT PROGRAM
        C:  X = 0
        T:  ENTER 3 NUMBERS, PRESS RETURN AFTER EACH
LOOP1   A:  #N
        J(N > X); LOOP2
        C:  X = N
LOOP2   J5:  EXIT1
        J:  LOOP1
EXIT1   T:  THE LARGEST NUMBER WAS #X
        E:
```

LOGO

The LOGO programming language, developed by Seymour Papert and his associates at MIT during the 1970s, is well-suited for teaching programming concepts to children. The language makes it easy for students to see how complex programs can be built by combining many smaller modules.

"Turtle graphics" is the most well-known aspect of the LOGO language. The programmer creates graphics by specifying instructions that move the cursor (which sometimes looks like a turtle) leaving a trail of lines. Figure 3.17 shows a screen display of a



superpilot

Makes it Easy!

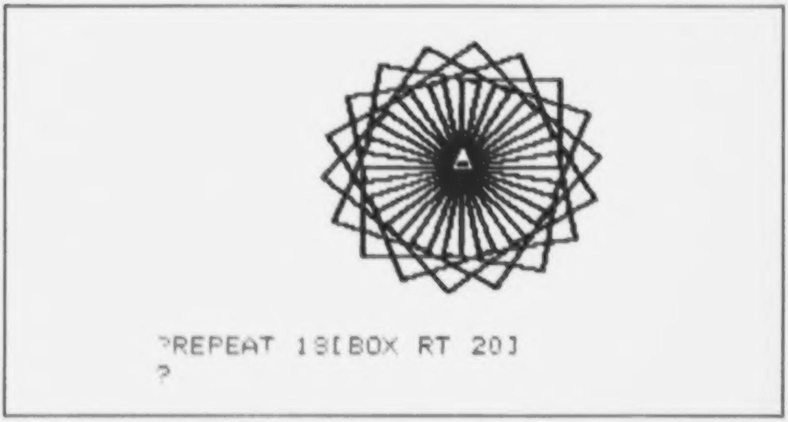
With features like these:

- Bold text
- Double-sized characters
- Color text and backgrounds
- Relative "turtle" graphics
- Animation
- Video disk control
- Record keeping

Let's take a look at them...

FIGURE 3.16

Reproduced with the permission of Apple Computer, Inc.



```
?REPEAT 18[BOX RT 20]
?
```

FIGURE 3.17

Reproduced with the permission of Apple Computer, Inc.

graphic created with the LOGO language. A sample LOGO program appears below.

```
TO BIGGEST
MAKE  "LARGEST 0
PRINT [ENTER A NUMBER WHEN THE QUESTION MARK APPEARS]
REPEAT 3 [MAKE "LARGEST BIG :LARGEST]
PRINT [THE LARGEST NUMBER WAS]
PRINT CHAR :LARGEST
END
```

Selecting a Programming Language

Which programming language should students use? Even computer experts do not agree which language is most suitable for students. The issue has been summarized by Osborne and Bunnell (1982) p. 81:

> The debate among software experts about which is the better language—Pascal, BASIC, or one of the many other languages—is unending, esoteric, and probably irrelevant. . . . Once you've learned one programming language, it will be easier to learn a second language. The most difficult aspect of programming is the concept of programming, not the particular language.

If you select a programming language for classroom use, you should carefully define why you are teaching programming. Computer programming is taught to students from kindergarten to the twelfth grade for a variety of reasons, including as part of computer literacy instruction, as college or employment preparation, and as an approach to learning how to solve problems.

You should also find out if your school or school district has already specified goals and objectives for computer programming and selected a particular language to accomplish those goals. Once you have established your objective(s) for teaching programming, you should become acquainted with using the programming language and teaching students to program.

CONCLUSION

As we have seen in this chapter, many tool programs can be used to support teachers and students in tasks related to instruction. This chapter was designed to help you realize that tool programs cannot be judged using generic evaluation criteria; they must be judged for their unique characteristics. Learn everything you can about each type of program and accept the fact that programs with numerous options will require more time to master.

READINGS AND RESOURCES

Garetz, Mark. *Bits, Bytes, and Buzzwords*. Dilithium Press, 8285 SW Nimbus Suite 151, Beaverton, OR 97005, 1983.

This book is an excellent resource if you are confused by all of the jargon associated with computers. Microcomputers, software, and peripherals are explained in a way that makes sense to the novice computer user.

Glossbrenner, Alfred. *How to Buy Software*. New York: St. Martin's Press, 1984.

This book is an excellent reference on how to judge the quality, price, and performance of a wide range of software.

Knapp, Linda. "Electronic Learning's Buyer's Guide: Word Processors." *Electronic Learning* 3 (1984):54–65.

This article includes software evalution suggestions as well as a ready-to-use software evaluation form for selecting word-processing programs.

The following products have been included as examples of educational software in this chapter.

1. *The Bank Street Writer* (Apple II+/IIe, Commodore 64, IBM PC, Atari 400/800)

Storytree (Apple II+/IIe, Commodore 64, IBM PC)

Publisher: Scholastic, Inc.
P.O. Box 7502
Jefferson City, MO 65102

2. *Apple Writer IIe* (Apple IIe)
Macwrite (MacIntosh)
Appleworks (Apple IIe/IIc)
Superpilot (Apple II+/IIe/IIc)
Apple Logo (Apple II+/IIe/IIc)
Publisher: Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014

3. *Report Card* (Apple II+/IIe)
Sensible Speller (Apple II+/IIe)
Publisher: Sensible Software
6619 Perham Drive
West Bloomfield, MI 48033

4. *The Print Shop* (Apple II+/IIe, Commodore 64)
Publisher: Broderbund Software
17 Paul Drive
San Rafael, CA 94903

5. *Crossword Magic* (Apple II+/IIe, Atari 400/800)
Publisher: L&S Computerware
1589 Fraser Drive
Sunnyvale, CA 94087

6. *PFS: Graph* (Apple II+/IIe)
Publisher: Software Publishing Corporation
1901 Landings Drive
Mt. View, CA 94043

7. *Songwriter* (Apple II+/IIe)
Publisher: Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591

8. *Koala Microillustrator* (Apple II+/IIe, Commodore 64, IBM PC)
Publisher: Koala Technologies
253 Marten Avenue
Mt. View, CA 94040

9. *Readability Tests* (Apple II+/IIe, IBM PC)
Word Find (Apple II+/IIe, IBM PC, Commodore 64, TRS Model III/4)
Publisher: Minneosta Educational Computing Consortium
3490 Lexington Avenue North
St. Paul, MN 55112

10. *Friendly Filer* (Apple IIe)
    Publisher: Grollier Electronic Publishing
    Sherman Turnpike
    Danbury, CN 06816

REFERENCES

Osborne, A., and Burnell, D. *An Introduction to Microcomputers*. **3d ed.** Berkeley, CA: McGraw-Hill, 1982.

PROJECTS AND QUESTIONS

1. Define each term listed below.
    a. word processing
    b. dedicated word processor
    c. 80-column program
    d. database
    e. spreadsheet program
    f. prompt
    g. icon
    h. integrated program
    i. BASIC
    j. LOGO
    k. PILOT
    l. Pascal
2. Examine a word-processing program displayed on a conventional television set, then examine the same program displayed on a computer monitor. Compare the results.
3. Examine three different word-processing programs. Rate each one on a scale from 1 to 5 (5 being easy to do) in the areas listed below. Compile the results into a comparison chart.
    a. Loading the program and beginning a new file.
    b. Moving the cursor.
    c. Correcting errors.
    d. Inserting text.
    e. Saving and retrieving a file.
    f. Printing a file.

4. List factors that might make it difficult to create a computer program that can read and criticize student compositions.
5. Predict five ways in which your role as a teacher might change if you or your students had unlimited access to tool programs.
6. Investigate three studies that have been conducted on the effects of using word processing with students. Include in your report a description of each study and the conclusions reached.
7. Select one of the following tool programs to complete a project listed below. Briefly summarize the program and your reaction to it.
    a. Use a word-processing program to write your summary.
    b. Use a database program to compile a list of thirty CAI programs.
    c. Use a graphing program to graph the prices of ten word-processing programs.
    d. Use a readability program to determine the reading level of this book.
    e. Use a puzzle-making program to create a puzzle that includes all of the terms listed in item one.
    f. Create a picture of a box using at least three of the programming languages discussed in this chapter.

4

Locating Educational Software

INTRODUCTION

There are between 30,000 and 40,000 software programs available today for personal computers. In the future there will be tens of thousands more. Some of these programs are appropriate for use by teachers and students—many are not. How will you begin to locate educational software? This chapter describes various sources of information about educational software.

BEGINNING THE SEARCH FOR SOFTWARE

Imagine that you have a $1,000 budget for locating and purchasing educational software for your class. Here are some procedures you might follow and traps to be avoided.

Define Your Needs and Resources

The first step is to think through why you want to acquire a particular program and how it will be integrated into your curriculum. Is your instructional goal to teach sixth grade students how to locate cities using latitude and longitude? Do you need a program that teaches French to high school students? Perhaps you want to obtain a program that allows you to use a single computer with the entire class.

You will need to know specific details about the hardware available at your school. If, for example, your school uses the Commodore 64 computer, you must seek software designed to run on a Commodore 64. You must also find out if your computer uses disks or tapes because programs available on disk require a disk drive and cannot be loaded into a computer via a cassette recorder.

What is the memory capacity of your computer? A 128K program will not run on a 64K machine. Some programs require the use of peripheral devices such as a color monitor, game paddles, or a printer. You must ensure that the program that appears to meet your needs also matches your hardware configuration.

Consider the Limitations of CAI

Once you determine your needs, think about the nature of the task you want to accomplish. Is your task within the capabilities of a microcomputer? Programs do not exist for every conceivable instructional task, so you should keep in mind that some subject matter may be taught better via conventional instructional methods. Pencils, paper, and books are still useful instructional tools.

Read Critical Reviews

Reading critical reviews written by educators for computer magazines can be an excellent source of information. A review can alert you to new software; however, a review is no substitute for trying out the software yourself. If several different reviewers rate a program highly, their recommendation should prompt you to review the program personally; but if a program receives several mediocre reviews, you may decide to spend your time examining alternative software.

To some extent, reviews are subjective and should be regarded as only one piece of information about a program. In fact, you should try to read several reviews of a product since reviewers rarely use identical evaluation criteria.

Software reviews appear in a number of computer magazines, some of which are written for parents or computer hobbyists. Reviews in magazines for a broad audience will be helpful, but educators are likely to obtain more detailed and relevant information about products by reading reviews in publications written specifically for educators. Figures 4.1 and 4.2 are examples of reviews from two popular computer magazines for educators, *The Computing Teacher* and *Electronic Learning*.

Some reviews provide comprehensive descriptions of a program, while others are more general. One highly regarded organization that provides comprehensive software reviews is MicroSIFT, whose review process results in a composite summary of evaluations made by at least two teachers who have experience at the subject and grade level of the material and one person who has experience with microcomputer software, hardware, and instructional applications. Figure 4.3 shows a MicroSIFT software review.

The Educational Products Information Exchange, (along with Consumers Union of the U.S., publisher of *Consumer Reports)* also publishes comprehensive software evaluations. The EPIE-CU evaluations—available in the form shown in Figures 4.4 and 4.5—are produced by a trained evaluation team.

Talk to Reliable Sources

"*Trigonometry for Tots* is a program every first-grade teacher should own!" exclaims Ms. Superlative. Should you run out and purchase the program on the basis of her recommendation?

Asking individuals who are knowledgeable about educational software to recommend programs can be a useful way to obtain information about software, but you should ask yourself the following questions:

FIGURE 4.1

Gertrude's Puzzles (1982)
The Learning Company, 4370 Alpine Road, Portola Valley, CA 94025
Target Age Group: 6 and up
Hardware Required: Apple II+, Apple IIe (with caps lock down), color monitor
Cost: $44.95
Additional Requirements: Joystick optional
Reviewed by Johnny W. Lott, University of Montana

Gertrude's Puzzles is a set of puzzles designed to help children develop reasoning skills. In these puzzles, players learn how to sort and classify objects. They also learn to solve problems with incomplete information and must analyze what they see. The program consists of two main sections. The first is a tutorial section which teaches the user how to pick up and drop shapes, control speed, and do all the things they need to know to move around in the world. The second section is the game part. Players solve puzzles by using loops (Venn diagrams), networks and boxes (which use classifying and sorting in a manner reminiscent of magic squares).

In all puzzles, players move pieces to form color and shape patterns. Some of the puzzles give players rules for playing, while others force players to determine the rules. When a puzzle is solved, Gertrude brings a treasure.

There are six different puzzle rooms—a simple and more advanced room for each of the three types of puzzles. In the network puzzles, misplaced pieces move slightly if placed incorrectly. In the loop puzzles, pieces fall out of the puzzle. Examples of completed puzzles are given in rooms that do not have doors—only viewing windows. The initial shapes used are geometric shapes, and another room is the Shape-Edit Room where players may create their own shapes.

Strong points: The software is attractive with very good graphics. Directions are very clear for players who are reading at fourth grade level. Some skill is required to use the control key to move a key to open the door to start the puzzles. The I, J, K and M keys are required to move the cursor up, down, right and left, and the repeat key will make the movement faster. A player can take as much time as needed. This software can be used by a player with little teacher direction.

Weak points: The reading level is too advanced for most six-year-olds. Average fourth graders thoroughly enjoyed it, as did talented seventh graders. In the software the impression is given that a joystick must be used in the Joystick Room, but that is not true. In the Blue Room of direction, the control key and space bar will not let the orange square be moved as the directions state, and there is a misspelled word. When first using the software additional problems emerge, which include the cursor not always picking up the piece that it is on, but another that is very close. In the Box puzzle, one piece is under another piece at the start of play. This can be very confusing to beginning players.

Recommendations:

Gertrude's Puzzles is a fascinating set of logic puzzles for both children and adults. It is as effective, or more, than using attribute pieces to teach classifying and sorting. It is too bad that the reading level is too high for the advertised age group, but otherwise it is an excellent piece of software.

[Ed. note: We have seen a final draft of the revised manual, and want readers to know that future editions will include an even more elaborate and helpful manual.]

Distributors:
Australia, Canada

Publisher's Reply:

Thank you for sending the proofs for the upcoming review of *Gertrude's Puzzles* for TLC comment. The educators within our Research and Development group have evaluated the issues the reviewer raised and have provided the following responses.

1. *Reading level too high for average six year old*—The age range six and up means six and up. Not all six year olds can be expected to be at the same skill level; however, some will be able to understand the reading/instructions.
2. *Orange square doesn't move*—The colored part of the cursor must touch it in order for it to work; however, in the new version this is not required.
3. *Misspelled word*—This is also fixed in the new version.

Again, thank you for allowing us to comment.

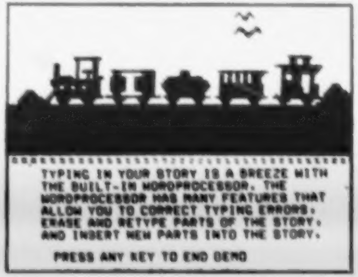
Karen Roebuck
Public Relations
The Learning Company

FIGURE 4.2

KIDWRITER

A creative writing program that helps young students write stories with words and pictures.

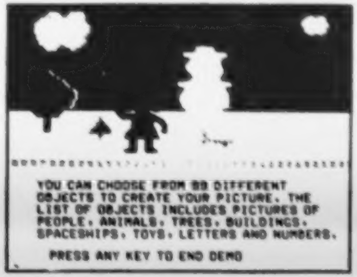
HARDWARE:	Apple II micros (48K) Commodore 64 (64K)
SOURCE:	Spinnaker Software 215 First St. Cambridge, MA 02142
PRICE:	$34.95
PREVIEW:	Demos available in retail stores
BACKUP:	Can be purchased for $12



TYPING IN YOUR STORY IS A BREEZE WITH THE BUILT-IN WORDPROCESSOR. THE WORDPROCESSOR HAS MANY FEATURES THAT ALLOW YOU TO CORRECT TYPING ERRORS, ERASE AND RETYPE PARTS OF THE STORY, AND INSERT NEW PARTS INTO THE STORY.

PRESS ANY KEY TO END DEMO

The Kidwriter *word processor*



YOU CAN CHOOSE FROM 99 DIFFERENT OBJECTS TO CREATE YOUR PICTURE. THE LIST OF OBJECTS INCLUDES PICTURES OF PEOPLE, ANIMALS, TREES, BUILDINGS, SPACESHIPS, TOYS, LETTERS AND NUMBERS.

PRESS ANY KEY TO END DEMO

Kidwriter *objects*

DESCRIPTION: SAY A STUDENT wants to write a story, but doesn't know *exactly* what to say. Spinnaker Software's *Kidwriter* provides one answer: this "inspirational" piece of software asks students to illustrate a story idea first, and *then* turn that story into words.

Kidwriter comes with a library of 99 characters and objects (people, animals, trees, spaceships, toys, trains, etc.) to choose from. Another library of scenes, from a farm to a theater to outer space, is available for the background.

To create a story, a student first selects options from the main menu to pick an object and a background scene to place it in. Then another menu appears, with options to change the color and size of the object, and to move it around on the screen.

When a student is satisfied with the picture, it's time to call up the word processor to change that image into words. The lower section of the screen displays text as the student types in a related story. The word processor has standard features such as the ability to erase and insert text, and correct typos.

When a picture-story page is finished, it can be saved on disk. Pages can also be linked together as a storybook. Over time, a whole collection of stories will be available for re-reading.

Applications: As an independent exercise in creative writing for grades 1-5 (though probably most appropriate in grades 2-4), this program would make a fine addition to any learning center or classroom.

Kidwriter does require teacher direction at first; but once familiar with the program commands, children are able to compose stories independently. Two or more students can also collaborate on stories.

Younger students may enjoy simply creating pictures, then dictating their stories to a teacher or older student to type in.

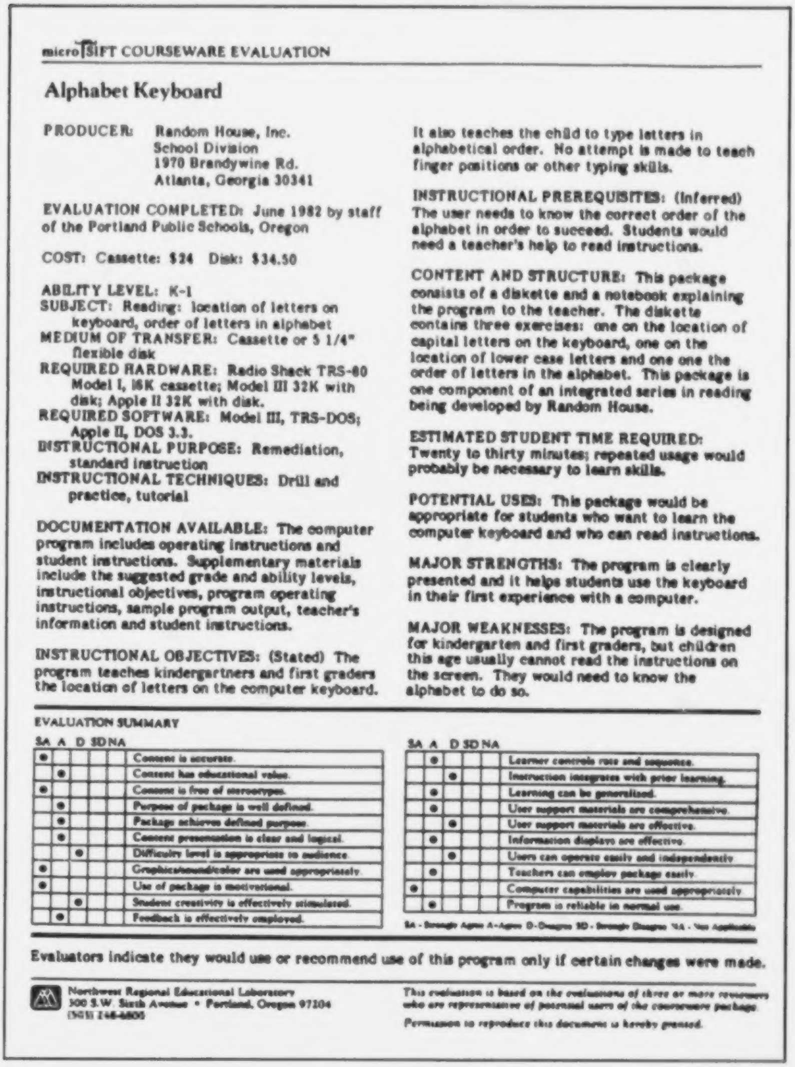
Comments: The process of first creating a picture, then describing it—the distinguishing feature of *Kidwriter*—is an excellent pre-writing alternative. The completed picture serves as a good story-starter, and provides a nice incentive for writing.

As an introduction to word processing, however, the program is less successful. Many of the children we watched using the program—particularly those who have no previous word processing experience—found it difficult to understand this component. Perhaps a built-in tutorial would be useful.

The instruction booklet contains a helpful list of the objects and characters available, though perhaps it would be even better if it were alphabetized. Some students also reported that they lost objects on the screen and found it hard to retrieve them.

With a little more attention to user instructions, *Kidwriter* could be a valuable introduction to word processing, as well as to writing. Despite the drawbacks in its execution, the program is based on an excellent concept, and seems well-received by children.

Reviewed by:
Joanne Marien, Principal
Nancy Radtke, 3rd Grade Teacher
Haldane Elementary
Cold Spring, NY



microSIFT COURSEWARE EVALUATION

Alphabet Keyboard

PRODUCER: Random House, Inc.
School Division
1970 Brandywine Rd.
Atlanta, Georgia 30341

EVALUATION COMPLETED: June 1982 by staff of the Portland Public Schools, Oregon

COST: Cassette: $24 Disk: $34.50

ABILITY LEVEL: K-1
SUBJECT: Reading: location of letters on keyboard, order of letters in alphabet
MEDIUM OF TRANSFER: Cassette or 5 1/4" flexible disk
REQUIRED HARDWARE: Radio Shack TRS-80 Model I, 16K cassette; Model III 32K with disk; Apple II 32K with disk.
REQUIRED SOFTWARE: Model III, TRS-DOS; Apple II, DOS 3.3.
INSTRUCTIONAL PURPOSE: Remediation, standard instruction
INSTRUCTIONAL TECHNIQUES: Drill and practice, tutorial

DOCUMENTATION AVAILABLE: The computer program includes operating instructions and student instructions. Supplementary materials include the suggested grade and ability levels, instructional objectives, program operating instructions, sample program output, teacher's information and student instructions.

INSTRUCTIONAL OBJECTIVES: (Stated) The program teaches kindergartners and first graders the location of letters on the computer keyboard. It also teaches the child to type letters in alphabetical order. No attempt is made to teach finger positions or other typing skills.

INSTRUCTIONAL PREREQUISITES: (Inferred) The user needs to know the correct order of the alphabet in order to succeed. Students would need a teacher's help to read instructions.

CONTENT AND STRUCTURE: This package consists of a diskette and a notebook explaining the program to the teacher. The diskette contains three exercises: one on the location of capital letters on the keyboard, one on the location of lower case letters and one one the order of letters in the alphabet. This package is one component of an integrated series in reading being developed by Random House.

ESTIMATED STUDENT TIME REQUIRED: Twenty to thirty minutes; repeated usage would probably be necessary to learn skills.

POTENTIAL USES: This package would be appropriate for students who want to learn the computer keyboard and who can read instructions.

MAJOR STRENGTHS: The program is clearly presented and it helps students use the keyboard in their first experience with a computer.

MAJOR WEAKNESSES: The program is designed for kindergarten and first graders, but children this age usually cannot read the instructions on the screen. They would need to know the alphabet to do so.

EVALUATION SUMMARY

SA	A	D	SD	NA	
●					Content is accurate.
	●				Content has educational value.
●					Content is free of stereotypes.
	●				Purpose of package is well defined.
	●				Package achieves defined purpose.
	●				Content presentation is clear and logical.
		●			Difficulty level is appropriate to audience.
●					Graphics/sound/color are used appropriately.
●					Use of package is motivational.
		●			Student creativity is effectively stimulated.
	●				Feedback is effectively employed.

SA	A	D	SD	NA	
	●				Learner controls rate and sequence.
		●			Instruction integrates with prior learning.
	●				Learning can be generalized.
	●				User support materials are comprehensive.
		●			User support materials are effective.
	●				Information displays are effective.
		●			Users can operate easily and independently.
	●				Teachers can employ package easily.
●					Computer capabilities are used appropriately.
	●				Program is reliable in normal use.

SA - Strongly Agree A - Agree D - Disagree SD - Strongly Disagree NA - Not Applicable

Evaluators indicate they would use or recommend use of this program only if certain changes were made.

Northwest Regional Educational Laboratory
300 S.W. Sixth Avenue • Portland, Oregon 97204
(503) 248-6800

This evaluation is based on the evaluations of three or more reviewers who are representative of potential users of the courseware package.

Permission to reproduce this document is hereby granted.

FIGURE 4.3

BEST COPY AVAILABLE

- Are his or her needs similar to your needs?
- Is he or she an experienced teacher and computer user?
- Does he or she know how to evaluate educational software?
- Can he or she competently answer your questions about a program?
- Will he or she demonstrate the program or lend it to you for review?

Read Product Announcements and Catalog Descriptions

Computer magazines and software catalogs contain advertising and product descriptions that can assist you in locating specific programs. It is important to remember that advertising claims may not be accurate and should never be regarded as a critical review of the product.

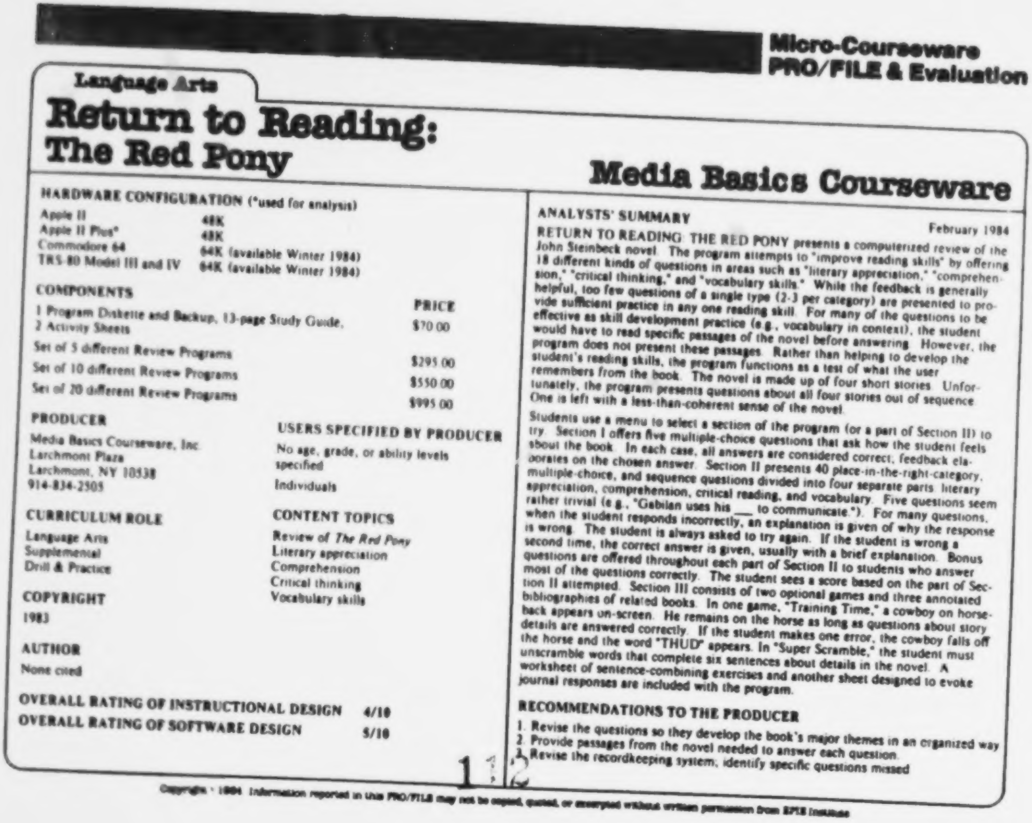
Examine Software Selection Guides

Many selection guides are available that summarize listings of educational software by subject area. Such guides can be used to help you determine whether or not a program exists for the subject area(s) that interests you.

TESS (*The Educational Software Selector*), a guide that lists over 4,200 educational software products, can tell you, for example, if there is a home economics software program for the IBM computer. A quick check of *TESS*'s "Summary Listings by Hardware System" provides program title, publisher's address, preview policy, phone number, grade level, scope, components, hardware configuration, and many other useful details.

Conduct an On-line Computer Search

If you have access to a computer equipped with a modem (a device used to communicate with other computers via the telephone), you can locate information about software by using a computerized information service. This approach is faster than trying to obtain and examine software reviews from catalogs and magazines.



Micro-Courseware
PRO/FILE & Evaluation

Language Arts

Return to Reading: The Red Pony

Media Basics Courseware

HARDWARE CONFIGURATION (*used for analysis)

Apple II	48K
Apple II Plus*	48K
Commodore 64	64K (available Winter 1984)
TRS-80 Model III and IV	64K (available Winter 1984)

COMPONENTS	PRICE
1 Program Diskette and Backup, 13-page Study Guide, 2 Activity Sheets	$70.00
Set of 5 different Review Programs	$295.00
Set of 10 different Review Programs	$550.00
Set of 20 different Review Programs	$995.00

PRODUCER

Media Basics Courseware, Inc.
Larchmont Plaza
Larchmont, NY 10538
914-834-2505

USERS SPECIFIED BY PRODUCER

No age, grade, or ability levels specified

Individuals

CURRICULUM ROLE

Language Arts
Supplemental
Drill & Practice

CONTENT TOPICS

Review of *The Red Pony*
Literary appreciation
Comprehension
Critical thinking
Vocabulary skills

COPYRIGHT

1983

AUTHOR

None cited

OVERALL RATING OF INSTRUCTIONAL DESIGN	**4/10**
OVERALL RATING OF SOFTWARE DESIGN	**5/10**

ANALYSTS' SUMMARY

February 1984

RETURN TO READING: THE RED PONY presents a computerized review of the
John Steinbeck novel. The program attempts to "improve reading skills" by offering
18 different kinds of questions in areas such as "literary appreciation," "comprehen-
sion," "critical thinking," and "vocabulary skills." While the feedback is generally
helpful, too few questions of a single type (2-3 per category) are presented to pro-
vide sufficient practice in any one reading skill. For many of the questions to be
effective as skill development practice (e.g., vocabulary in context), the student
would have to read specific passages of the novel before answering. However, the
program does not present these passages. Rather than helping to develop the
student's reading skills, the program functions as a test of what the user
remembers from the book. The novel is made up of four short stories. Unfor-
tunately, the program presents questions about all four stories out of sequence.
One is left with a less-than-coherent sense of the novel.

Students use a menu to select a section of the program (or a part of Section II) to
try. Section I offers five multiple-choice questions that ask how the student feels
about the book. In each case, all answers are considered correct; feedback ela-
borates on the chosen answer. Section II presents 40 place-in-the-right-category,
multiple-choice, and sequence questions divided into four separate parts: literary
appreciation, comprehension, critical reading, and vocabulary. Five questions seem
rather trivial (e.g., "Gabilan uses his ___ to communicate."). For many questions,
when the student responds incorrectly, an explanation is given of why the response
is wrong. The student is always asked to try again. If the student is wrong a
second time, the correct answer is given, usually with a brief explanation. Bonus
questions are offered throughout each part of Section II to students who answer
most of the questions correctly. The student sees a score based on the part of Sec-
tion II attempted. Section III consists of two optional games and three annotated
bibliographies of related books. In one game, "Training Time," a cowboy on horse-
back appears on-screen. He remains on the horse as long as questions about story
details are answered correctly. If the student makes one error, the cowboy falls off
the horse and the word "THUD" appears. In "Super Scramble," the student must
unscramble words that complete six sentences about details in the novel. A
worksheet of sentence-combining exercises and another sheet designed to evoke
journal responses are included with the program.

RECOMMENDATIONS TO THE PRODUCER

1. Revise the questions so they develop the book's major themes in an organized way.
2. Provide passages from the novel needed to answer each question.
3. Revise the recordkeeping system; identify specific questions missed.

Copyright © 1984. Information reported in this PRO/FILE may not be copied, quoted, or excerpted without written permission from EPIE Institute.

FIGURE 4.4

EPIE Institute and Consumers Union Micro-Courseware PRO/FILEs and Evaluations, P.O. Box 839, Watermill, NY 11976

BEST COPY AVAILABLE

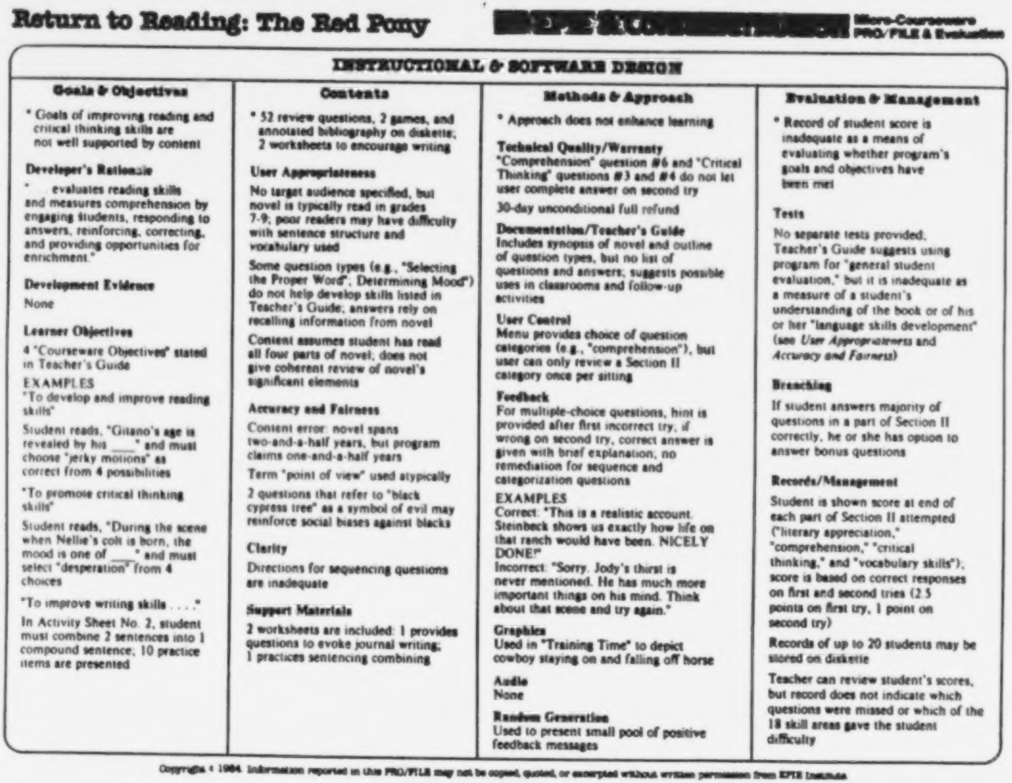


FIGURE 4.5

Return to Reading: The Red Pony

Micro-Courseware
PRO/FILE & Evaluation

INSTRUCTIONAL & SOFTWARE DESIGN

Goals & Objectives

- Goals of improving reading and critical thinking skills are not well supported by content

Developer's Rationale

" . . . evaluates reading skills and measures comprehension by engaging students, responding to answers, reinforcing, correcting, and providing opportunities for enrichment."

Development Evidence

None

Learner Objectives

4 "Courseware Objectives" stated in Teacher's Guide

EXAMPLES
"To develop and improve reading skills"

Student reads, "Gitano's age is revealed by his ____" and must choose "jerky motions" as correct from 4 possibilities

"To promote critical thinking skills"

Student reads, "During the scene when Nellie's colt is born, the mood is one of ____" and must select "desperation" from 4 choices

"To improve writing skills . . . ."

In Activity Sheet No. 2, student must combine 2 sentences into 1 compound sentence; 10 practice items are presented

Contents

- 52 review questions, 2 games, and annotated bibliography on diskette; 2 worksheets to encourage writing

User Appropriateness

No target audience specified, but novel is typically read in grades 7-9; poor readers may have difficulty with sentence structure and vocabulary used

Some question types (e.g., "Selecting the Proper Word", Determining Mood") do not help develop skills listed in Teacher's Guide; answers rely on recalling information from novel

Content assumes student has read all four parts of novel; does not give coherent review of novel's significant elements

Accuracy and Fairness

Content error: novel spans two-and-a-half years, but program claims one-and-a-half years

Term "point of view" used atypically

2 questions that refer to "black cypress tree" as a symbol of evil may reinforce social biases against blacks

Clarity

Directions for sequencing questions are inadequate

Support Materials

2 worksheets are included: 1 provides questions to evoke journal writing; 1 practices sentencing combining

Methods & Approach

- Approach does not enhance learning

Technical Quality/Warranty
"Comprehension" question #6 and "Critical Thinking" questions #3 and #4 do not let user complete answer on second try

30-day unconditional full refund

Documentation/Teacher's Guide
Includes synopsis of novel and outline of question types, but no list of questions and answers; suggests possible uses in classrooms and follow-up activities

User Control
Menu provides choice of question categories (e.g., "comprehension"), but user can only review a Section II category once per sitting

Feedback
For multiple-choice questions, hint is provided after first incorrect try; if wrong on second try, correct answer is given with brief explanation; no remediation for sequence and categorization questions

EXAMPLES
Correct: "This is a realistic account. Steinbeck shows us exactly how life on that ranch would have been. NICELY DONE!"
Incorrect: "Sorry. Jody's thirst is never mentioned. He has much more important things on his mind. Think about that scene and try again."

Graphics
Used in "Training Time" to depict cowboy staying on and falling off horse

Audio
None

Random Generation
Used to present small pool of positive feedback messages

Evaluation & Management

- Record of student score is inadequate as a means of evaluating whether program's goals and objectives have been met

Tests

No separate tests provided; Teacher's Guide suggests using program for "general student evaluation," but it is inadequate as a measure of a student's understanding of the book or of his or her "language skills development" (see *User Appropriateness* and *Accuracy and Fairness*)

Branching

If student answers majority of questions in a part of Section II correctly, he or she has option to answer bonus questions

Records/Management

Student is shown score at end of each part of Section II attempted ("literary appreciation," "comprehension," "critical thinking," and "vocabulary skills"), score is based on correct responses on first and second tries (2.5 points on first try, 1 point on second try)

Records of up to 20 students may be stored on diskette

Teacher can review student's scores, but record does not indicate which questions were missed or which of the 18 skill areas gave the student difficulty

Copyright © 1984. Information reported in this PRO/FILE may not be copied, quoted, or excerpted without written permission from EPIE Institute.

BEST COPY AVAILABLE

RICE (Resources In Computer Education) is a computerized information service used by some educators to locate software. It is a comprehensive database of information about several thousand microcomputer programs for use in elementary and secondary education. RICE (and a number of other databases) will provide software reviews and titles of educational programs for a fee.

Attend Professional Conferences

Computers-in-education conferences occur almost weekly throughout the United States, and most of them include vendor exhibits where software producers display their programs. Many conferences include presentations that focus on existing software or programs about to be released.

Locate More Than One Program

If possible, you should try to locate titles and reviews for at least three programs that appear to meet the needs you have defined. You should try to preview each of these programs so that you can compare them.

Comparison shopping for software is a good idea for several reasons. First, it can help you determine a baseline for software prices and performance. Second, it will increase your confidence that no less expensive or better quality program was available.

OBTAINING SOFTWARE TO PREVIEW

Previewing software prior to purchase is an *absolute* necessity; however, obtaining software to preview is not always easy. Many software producers are reluctant to provide programs for preview because they fear their programs will be illegally copied. The following sections describe different ways of obtaining software to preview.

Write to the Publisher

When you locate a software title that interests you, contact the publisher or sales representative and request a preview copy of the program. Some software vendors will provide materials on thirty-day approval. This means that you have thirty days to examine the program and then decide whether to purchase it or return it to the publisher.

If the publisher will not provide a program for preview, request the address of a school near you that has purchased the program. You may be able to contact the school and arrange to preview their software.

Visit Local Computer Stores

Stores that sell software sometimes allow customers to preview their programs. Unfortunately, most computer stores do not stock a large selection of educational software.

Visit a Preview Center

An excellent way to examine software is to visit a software preview center. Regional centers—sponsored by education agencies, colleges, teacher centers, and computer dealers—exist across the United States for the purpose of providing educators with an opportunity to examine programs without paying a fee.

Free and Inexpensive Software

Independent programmers and amateur software authors will sometimes offer their programs to the general public at little or no cost. These programmers offer their work either to showcase their achievement or to help other people solve a common problem. Programs in this category are called "public domain software." You may discover a "pearl" among public domain software programs, but many of them contain errors and are instructionally mediocre.

CONCLUSION

Locating software requires knowledge, organization, persistence, and luck. If you can't locate software for a particular purpose, don't despair; educational software production is a new and rapidly developing field. Today's needs are tomorrow's programs.

Remember that defining your software needs and locating programs to preview constitute a preliminary step in software selection. The next step is establishing the criteria and process that you will use to evaluate a program.

READINGS AND RESOURCES

Brown, Steven; Grossman, George C.; and Polson, Nicola. "Educational Software Reviews: Where are They?" *The Computing Teacher* 12 (1984): 33–37.

This article presents a comprehensive summary of publications that review educational software.

Classroom Computer Learning Directory of Educational Computing Resources. Dayton, OH: Classroom Computer Learning.

This annual publication provides a comprehensive collection of up-to-date information about computer education resources.

"EL's National Directory of Software Preview Centers: Part I." *Electronic Learning* 3 no. 4 (1984):59.

"EL's National Directory of Software Preview Centers: Part II." *Electronic Learning* 3 no. 5 (1984):68.

These articles include the address of software preview centers for every state in the United States.

"Public Domain Software Listings." *The Computing Teacher* 11 (1984): 46–59.

This article lists a number of public domain software titles and sources.

Publishers Offering Previews

The following publishers have thirty-day preview policies. You should inquire about the details of their policies and determine if such policies are still in force.

Ahead Designs
699 N. Vulcan #88
Encinatas, CA 92024

Applecart
515 N. Franklin
Juneau, AK 99801

Aquarius
P.O. Box 128
Indian Rocks Beach, FL 33535

AU Software
P.O. Box 597
Colleyville, TX 76034

BCD Associates, Inc.
5807 S. W. 5th Suite 101
Oklahoma City, OK 73128

Bluebirds Computer Software
P.O. Box 339
Wyandotte, MI 48192

The Cactusplot Company
1442 North McAllister
Tempe, AZ 85281

Calico, Inc.
P.O. Box 15916
St. Louis, MO 63114

CIMS
Prince Laboratory School
University of Northern Iowa
Cedar Falls, IA 50613

Compu-Tations
P.O. Box 502
Troy, MI 48099

Computer Scholar, Inc.
820 California St.
Oceanside, CA 92054

Cow Bay Computing
Box 515
Manhasset, NY 11030

Creative Publications
P.O. Box 10328
Palo Alto, CA 94303

Cuisenaire Company of America
12 Church St. Box D
New Rochelle, NY 10805

D&M Software Publishers
1510 S. 97th St.
Tacoma, WA 98444

Diversified Education Enterprises
725 Main St.
Lafayette, IN 47901

Earthware Computer Services
P.O. Box 30039
Eugene, OR 97403

Educational Courseware
3 Napa Lane
Westport, CT 06880

Educational Materials and Equipment
P.O. Box 17
Pelham, NY 10803

Educational Performance Systems
309 Office Plaza Drive
POB 12951 Suite 204
Tallahassee, FL 32308

Educational Software
P.O. Box 746
McMinville, OR 97128

EduTech, Inc.
634 Commonwealth Ave.
Newton Centre, MA 02159

Electronic Courseware System
309 Windsor Road
Champaign, IL 61820

Encyclopedia Brittanica
Educational Corporation
425 North Michigan Ave.
Chicago, IL 60611

Gessler Publishing Company
900 Broadway
New York, NY 10003

Gregg-McGraw-Hill
1221 Avenue of the Americas
New York, NY 10020

J. L. Hammett Company
Microcomputer Division
Hammett Place
Braintree, MA 02184

HEI, Inc.
1495 Steiger Lake Lane
Victoria, MN 55386

Island Software
Box 300 Dept. B
Lake Grove, NY 11755

JMH Software of Minnesota
P.O. Box 41308
Minneapolis, MN 55441

K-12 Micromedia
172 Broadway
Woodcliff Lake, NJ 07675

Laureate Learning Systems
One Mill Street
Burlington, VT 05401

Linguametrics Group
P.O. Box 3495
San Rafael, CA 94912–3495

McNeil Associates
101 Glendale Road
Exton, PA 19341

MECC Publications Dept.
2520 Broadway Drive
St. Paul, MN 55113

Melcher Software
412 Hollybrook Drive
Midland, MI 48640

Merry Bee Communications
815 Crest Drive
Omaha, NE 68046

Microcomputer Workshops Corporation
225 Westchester Avenue
Portchester, NY 10573

Micro Learningware
P.O. Box 307
North Mankato, MI 56001

Milliken Publishing Company
1100 Research Blvd
St. Louis, MO 63132

Mount Castor Industries
368 Shays Street
Amherst, MA 01002

Polytech
428 North Broadway
Peru, IN 46970

Programs For Learning, Inc.
P.O. Box 954
New Milford, CN 06776

Redcomp Services
624 West Chenango Road
Castle Creek, NY 13744

River Bend Software
P.O. Box 637
Atchison, KS 66002–0637

Scholastic, Inc.
902 Sylvan Ave.
Englewood Cliffs, NJ 07632

Sonora Enterprises
P.O. Box 4841
Albuquerque, NM 87196

South Micro Systems For Educators
716 East Davis Street
Burlington, NC 27215

Spectrum Software
75 D. Todd Pond Road
Lincoln, MA 01773

Sterling Swift Publishing Company
7901 South IH–35
Austin, TX 78744

Sunburst Communications, Inc.
39 Washington Avenue Room VF 414
Pleasantville, NY 10570

Systems Design Association, Inc.
723 Kanawha Blvd. East
Charleston, WV 25301

TIES
1925 West County Road B2
St. Paul, MN 55113

Universal Systems For Education
14901 East Hamden Avenue #250
Aurora, CO 80014

J. W. Walsh Publisher
321 Valley Street
Portland, ME 04102

The Wizard
18584 Carlywn Drive
Castro Valley, CA 94546

Zeitgeist
5150 North 6th Street
Suite 179
Fresno, CA 93710

Sources of Educational Software Reviews

Educational Computing Journals

AEDS Monitor
Association for Educational Data Systems

1201 Sixteenth St., N. W.
Washington, DC 20036
Average number of reviews per issue: 0–2

Classroom Computer Learning
2451 E. River Rd.
Dayton, OH 45439
Average number of reviews per issue: 4

Computers, Reading and Language Arts
P.O. Box 13247
Oakland, CA 94661
Average number of reviews per issue: 8

Computers in the Schools
The Haworth Press, Inc.
28 E. 22nd St.
New York, NY 10010
Average number of reviews per issue: Varies

The Computing Teacher
University of Oregon
Eugene, OR 97403
Average number of reviews per issue: 5

Educational Technology
140 Sylvan Ave.
Englewood Cliffs, NJ 07632
Average number of reviews per issue: 4

Electronic Education
Suite 220
1311 Executive Center Drive
Tallahassee, FL 32301
Average number of reviews per issue: 1

Journal of Computers in Mathematics and Science Teaching
Box 4455
Austin, TX 78765
Average number of reviews per issue: 10

Mathematics and Computer Education
Old Bethpage Rd.
Long Island, NY 11804
Average number of reviews per issue: 3–6

Teaching and Computers
902 Sylvan Avenue
Englewood Cliffs, NJ 07632
Average number of reviews per issue: 7

Review Journals

The Apple Journal of Courseware Review
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
Average number of reviews per issue: 15

Digest of Software Reviews
301 West Mesa
Fresno, CA 93704
Average number of reviews per issue: 50

Evaluations: Microware
7351 Elmbridge Way
Richmond, B.C., Canada V6X 1B8
Average number of reviews per issue: 25

Hively's Choice
520 E. Bainbridge St.
Elizabethtown, PA 17022-9989
Average number of reviews per issue: 106

Library Software Review
520 Riverside Ave.
Westport, CT 06880
Average number of reviews per issue: Varies

MicroSIFT Reviews
Northwest Regional Educational Laboratory
300 S. W. Sixth Avenue
Portland, OR 97204
Average number of reviews per issue: 100 per year

SECTOR PROJECT
(Special Education Computer Technology Online Resources)
Exceptional Child Center, UMC-68
Utah State University
Logan, UT 84322
Average number of reviews per issue: 58

Software Reports
2101 Las Palmas Dr.
Carlsbad, CA 92008
Average number of reviews per issue: 382 pages of reviews anually

Education Periodicals

Arithmetic Teacher
1906 Association Dr.
Reston, VA 22091
Average number of reviews per issue: 3

Booklist
50 East Huron St.
Chicago, IL 60611
Average number of reviews per issue: Varies

Curriculum Review
517 S. Jefferson St.
Chicago, IL 60607
Average number of reviews per issue: 10

Instructor
757 Third Ave.
New York, NY 10017
Average number of reviews per issue: 0–2

Journal of Learning Disabilities
11 E. Adams St.
Chicago, IL 60603

Mathematics Teacher
1906 Association Dr.
Reston, VA 22091
Average number of reviews per issue: 7

Media and Methods
1511 Walnut St.
Philadelphia, PA 19102
Average number of reviews per issue: 2

Physics Teacher
Graduate Physics Building, SUNY
Stony Brook, NY 11794
Average number of reviews per issue: 1

School Science and Mathematics
Bowling Green State University
126 Life Science Building
Bowling Green, OH 43403

Average number of reviews per issue: 2

Educational Computing Newsletters

CHIME (Clearinghouse of Information on Microcomputers in Education)
108 Gundersun, Oklahoma State University
Stillwater, OK 74078

Average number of reviews per issue: 10

Closing The Gap
P.O. Box 68
Henderson, MN 56044

Average number of reviews per issue: 2

CUE (Computer-Using Educators)
127 O'Connor St.
Menlo Park, CA 94025

Average number of reviews per issue: 3–4

Microcomputers In Education
5 Chapel Hill Dr.
Fairfield, CT 06432

Average number of reviews per issue: 6

Users: The MECC Instructional Computing Newsletter
3490 Lexington North
St. Paul, MN 55112

Other Sources of Information

PRO/FILES
A box with several hundred in-depth evaluations, with a minimum of forty updates every two months. Each evaluation, conducted by EPIE's network of evaluator's, is four to six pages and includes screen displays, an overall rating, student comments, and teacher use. Available from:

EPIE Institute
P.O. Box 839
Water Mill, NY 11976

The Educational Software Selector (TESS)
A comprehensive directory and evaluation guide containing detailed

information on over 5,000 pieces of educational software. The guide includes review ratings and refers to reviews in fourteen other sources. Available from:

EPIE Institute
P.O. Box 839
Water Mill, NY 11976

PROJECTS AND QUESTIONS

1. Read three software reviews, each from a different educational computing journal. Summarize the reviewers' comments and your reaction to the information presented in each review.
2. Locate two reviews for the same program and compare the comments made by each reviewer. Make a list of program features that each reviewer discussed.
3. Visit a computer store or software dealer in your area. Describe your visit. Issues to address are suggested below.
   a. How many educational programs does the store stock?
   b. Can software be previewed on site?
   c. Can you borrow programs to preview at home?
   d. Is the sales staff knowledgeable about educational software?
   e. How do the software prices compare with prices listed in catalogs?
4. Locate and examine ten educational software catalogs. Make a list of software distributors that allow consumers to preview software prior to purchase.
5. Make a notebook of educational software advertisements. Analyze and explain the advertising techniques used to sell each program.
6. Create a plan for organizing a software "Preview Day" in a school or district. Select a site, target specific programs to be previewed, list the names of local vendors who might

participate, and describe organizational strategies that would permit a large number of educators to preview the programs.

7. Interview five teachers who use educational software on a regular basis. Ask each teacher how he or she locates programs to use in the classroom. Summarize the results.

5

Conducting Software Evaluations and Reviews

INTRODUCTION

This chapter presents information about how to evaluate and review eductional software and suggests a number of questions that may be asked about educational software. Educators must recognize that there is a difference between evaluating software and reviewing software. Evaluating usually involves many hours, people, and details about a program.

> Dr. Hal Peters of CONDUIT estimates that the review and evaluation of a typical program requires approximately 40 hours of the time of content and programming specialists (Gleason, 1981, p. 13).

Reviews require less in terms of personnel, time, and attention to detail and may be the result of a comprehensive evaluation. Both evaluations and reviews are important in the process of selecting software.

SELECTION CRITERIA AND EVALUATION FORMS

Before you evaluate or review a program, you must have a set of questions to ask about a program. Since evaluation implies that standards exist, evaluating software requires that you know the characteristics of a high-quality program. The problem is that educators do not always agree about what constitutes quality software; therefore, software evaluation is very much like judging wine or reviewing a movie: it is susceptible to the evaluator's subjective judgment.

MicroSIFT, EPIE, and other groups have tried to make software evaluation an objective process by devising comprehensive evaluation forms and systems for summarizing an evaluator's responses. No single evaluation system is satisfactory for all categories of educational software. Rapid changes in the field also make it difficult to specify criteria that reflect the latest improvements in software design and computer technology.

The design of an evaluation form can influence how an evaluator addresses specific questions about a program. The evaluator's judgment can be affected if an evaluation form uses vague terms or presents too many questions. Consider your personal preference about forms: Do lengthy forms intimidate you? Do you prefer a checklist or are you more comfortable writing comments? Does the form ask the same questions that you would ask about a program?

An evaluation form can be a useful tool for providing focus as well as a procedure for examining programs. One excellent evaluation instrument is the *MicroSIFT Evaluator's Guide for Microcomputer-Based Instructional Packages*. This guide provides clearly defined evaluation criteria and well-organized evaluation forms. Figures 5.1 through 5.3 show examples of completed MicroSIFT evaluation forms.

THE EVALUATION AND REVIEW PROCESS

Preparing to Evaluate Educational Software

If you plan to conduct a software evaluation you should consider making the process a cooperative effort. An evaluation committee should include several teachers with experience in teaching the subject material and the grade level for which a program is intended. It is also helpful to have one committee member who is knowledgeable about evaluating instructional software; this individual can serve as the committee's facilitator and synthesize the evaluation results.

An evaluation committee should become familiar with the evaluation process before examining a program. The committee members should read the evaluation guide chosen, discuss the meaning of the criteria, and review samples of completed evaluation forms. A useful learning procedure is to evaluate a program with the guidance of an experienced software evaluator. This procedure allows committee members to raise questions and discuss specific details about a program with a knowledgeable trainer.

Software Evaluation Activities

The following sequence of activities is typical of many software evaluation processes.

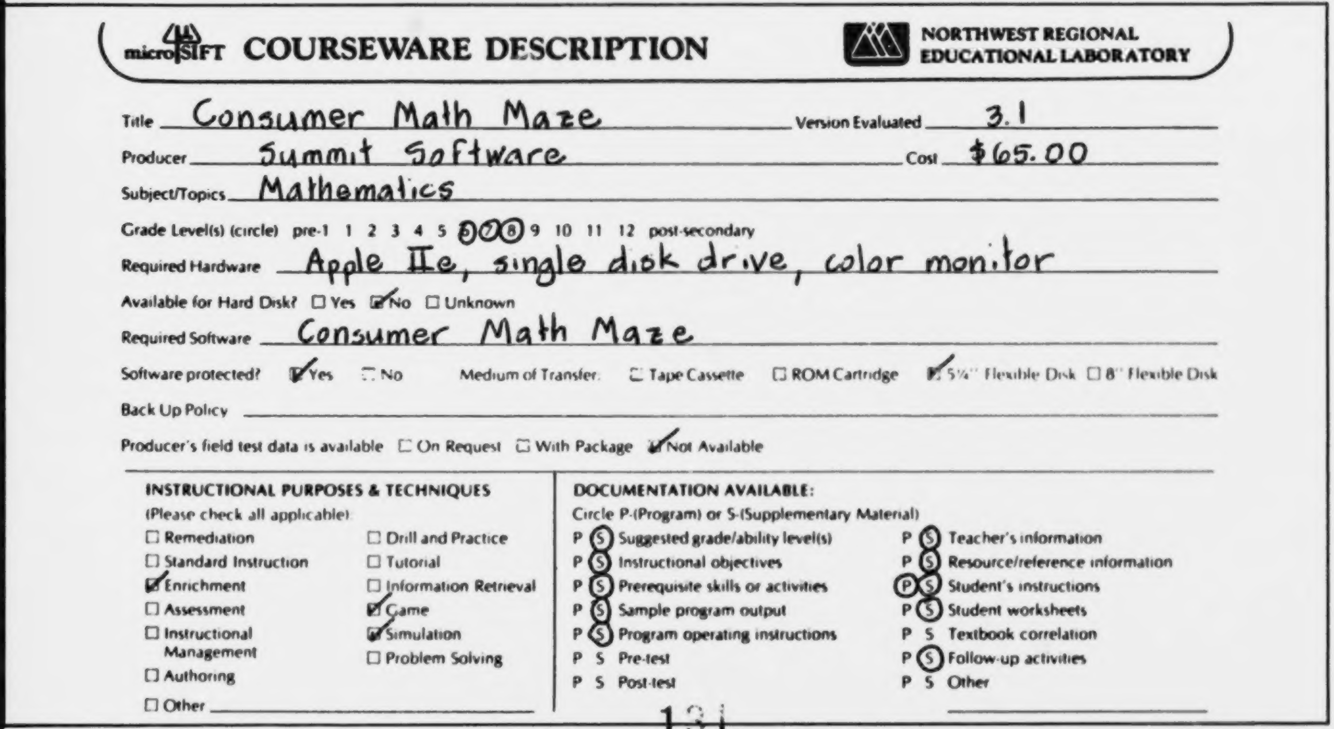
Program Screening and Selection. A knowledgeable individual uses a screening process in order to eliminate programs of inferior quality. The program's operational readiness and hardware compatability can also be determined.

Documentation Review and Program Description. The evaluation committee should begin its evaluation by carefully examining the program's manuals and support materials. An evaluation form can be used to identify and record the factual information about using the program.

Program Familiarization. The committee should actually run the program to become familar with its basic operation.

Program Testing. The committee must examine the program carefully and consider a number of evaluation questions related to the program's operational and instructional design. This information should also be recorded on the evaluation form.

FIGURE 5.1



microSIFT COURSEWARE DESCRIPTION

NORTHWEST REGIONAL EDUCATIONAL LABORATORY

Title Consumer Math Maze Version Evaluated 3.1

Producer Summit Software Cost $65.00

Subject/Topics Mathematics

Grade Level(s) (circle) pre-1 1 2 3 4 5 [✓ 6] [✓ 7] [✓ 8] 9 10 11 12 post-secondary

Required Hardware Apple IIe, single disk drive, color monitor

Available for Hard Disk? ☐ Yes ☑ No ☐ Unknown

Required Software Consumer Math Maze

Software protected? ☑ Yes ☐ No Medium of Transfer: ☐ Tape Cassette ☐ ROM Cartridge ☑ 5¼" Flexible Disk ☐ 8" Flexible Disk

Back Up Policy

Producer's field test data is available ☐ On Request ☐ With Package ☑ Not Available

INSTRUCTIONAL PURPOSES & TECHNIQUES

(Please check all applicable)

☐ Remediation
☐ Standard Instruction
☑ Enrichment
☐ Assessment
☐ Instructional Management
☐ Authoring
☐ Other

☐ Drill and Practice
☐ Tutorial
☐ Information Retrieval
☑ Game
☑ Simulation
☐ Problem Solving

DOCUMENTATION AVAILABLE:

Circle P-(Program) or S-(Supplementary Material)

P [✓ S] Suggested grade/ability level(s)
P [✓ S] Instructional objectives
P [✓ S] Prerequisite skills or activities
P [✓ S] Sample program output
P [✓ S] Program operating instructions
P S Pre-test
P S Post-test

P [✓ S] Teacher's information
P [✓ S] Resource/reference information
[✓ P] [✓ S] Student's instructions
P [✓ S] Student worksheets
P S Textbook correlation
P [✓ S] Follow-up activities
P S Other

BEST COPY AVAILABLE

OBJECTIVES: ☑ Stated ☐ Inferred

To provide students with opportunities to solve consumer-math type story problems.

PREREQUISITES: ☐ Stated ☑ Inferred

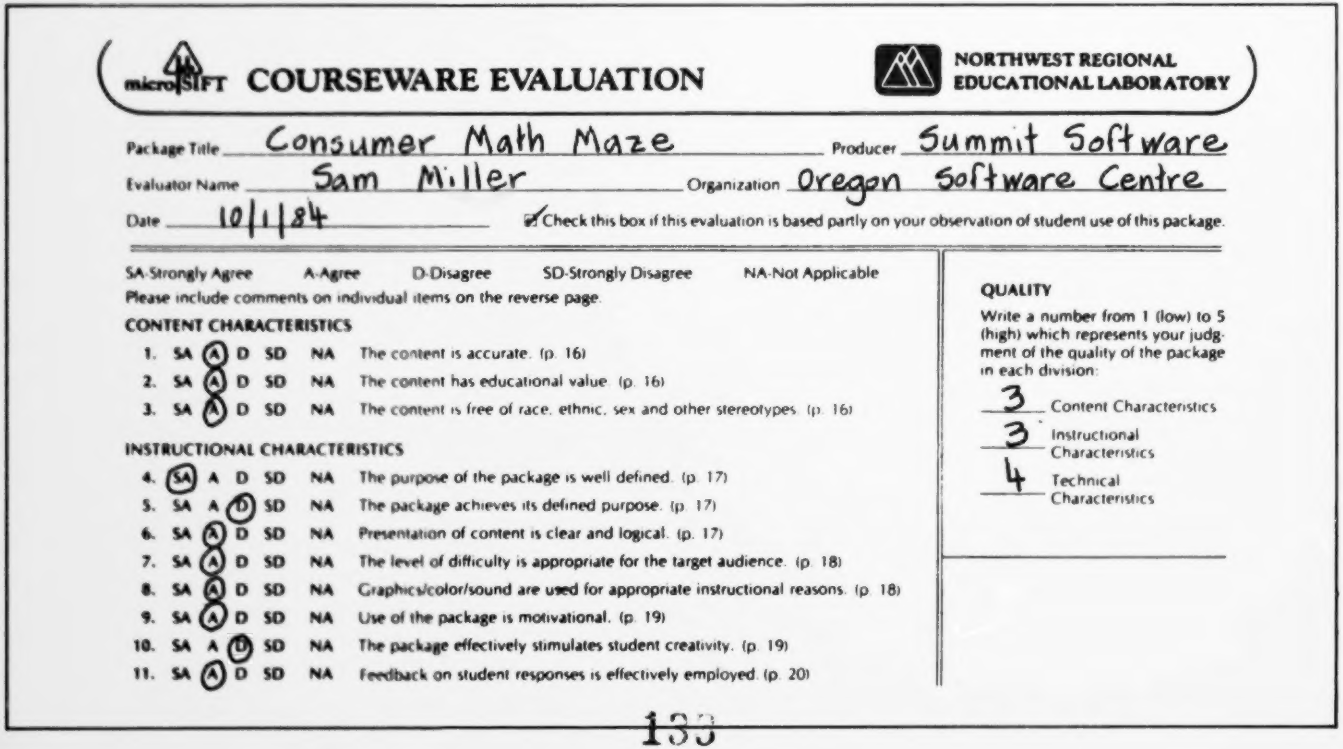
Assumes that students can read at an adult reading level. Assumes that students have knowledge of fractions, percents and basic operations in math.

Describe package **CONTENT AND STRUCTURE,** including record keeping and reporting functions:

This is a simulation game of non-typical consumer math problems. Students are asked to believe that they have unlimited funds to spend in a department store. Students negotiate the department store "maze" by answering math problems correctly.

(Use back if additional space needed)

FIGURE 5.2



microSIFT COURSEWARE EVALUATION

NORTHWEST REGIONAL EDUCATIONAL LABORATORY

Package Title Consumer Math Maze Producer Summit Software

Evaluator Name Sam Miller Organization Oregon Software Centre

Date 10/1/84 ☑ Check this box if this evaluation is based partly on your observation of student use of this package.

SA-Strongly Agree A-Agree D-Disagree SD-Strongly Disagree NA-Not Applicable

Please include comments on individual items on the reverse page.

CONTENT CHARACTERISTICS

1. SA [✓ A] D SD NA The content is accurate. (p. 16)
2. SA [✓ A] D SD NA The content has educational value. (p. 16)
3. SA [✓ A] D SD NA The content is free of race, ethnic, sex and other stereotypes. (p. 16)

INSTRUCTIONAL CHARACTERISTICS

4. [✓ SA] A D SD NA The purpose of the package is well defined. (p. 17)
5. SA A [✓ D] SD NA The package achieves its defined purpose. (p. 17)
6. SA [✓ A] D SD NA Presentation of content is clear and logical. (p. 17)
7. SA [✓ A] D SD NA The level of difficulty is appropriate for the target audience. (p. 18)
8. SA [✓ A] D SD NA Graphics/color/sound are used for appropriate instructional reasons. (p. 18)
9. SA [✓ A] D SD NA Use of the package is motivational. (p. 19)
10. SA A [✓ D] SD NA The package effectively stimulates student creativity. (p. 19)
11. SA [✓ A] D SD NA Feedback on student responses is effectively employed. (p. 20)

QUALITY

Write a number from 1 (low) to 5 (high) which represents your judgment of the quality of the package in each division:

3 Content Characteristics

3 Instructional Characteristics

4 Technical Characteristics

12. SA A D SD [✓ NA] The learner controls the rate and sequence of presentation and review. (p. 20)
13. SA A D SD [✓ NA] Instruction is integrated with previous student experience. (p. 29)
14. SA A [✓ D] SD NA Learning can be generalized to an appropriate range of situations. (p. 29)

TECHNICAL CHARACTERISTICS

15. SA [✓ A] D SD NA The user support materials are comprehensive. (p. 30)
16. SA [✓ A] D SD NA The user support materials are effective. (p. 31)
17. SA A [✓ D] SD NA Information displays are effective. (p. 31)
18. SA A [✓ D] SD NA Intended users can easily and independently operate the program. (p. 32)
19. SA [✓ A] D SD NA Teachers can easily employ the package. (p. 34)
20. SA [✓ A] D SD NA The program appropriately uses relevant computer capabilities. (p. 34)
21. [✓ SA] A D SD NA The program is reliable in normal use. (p. 35)

RECOMMENDATIONS

☐ I highly recommend this package.

☐ I would use or recommend use of this package with little or no change. (Note suggestions for effective use below.)

☑ I would use or recommend use of this package only if certain changes were made. (Note changes under Weaknesses or Other Comments.)

☐ I would not use or recommend this package. (Note reasons under Weaknesses.)

Describe the potential use of the package in classroom settings:

Possibly useful for enrichment math activities with a group or an individual student. Might serve as a recreational activity for students who have mastered basic computation skills and who demonstrate the ability to read, understand and solve story problems.

FIGURE 5.3

Estimate the amount of time a student would need to work with the package in order to achieve the objectives:
(Can be total time, time per day, time range or other indicator.)

25-30 minutes to become facile with the program.

May require the presence of a teacher to introduce the program to some groups of students.

Strengths

The activities have the potential to be a humorous and entertaining alternative for practicing story problems.

Weaknesses:

Reading level of the program is beyond the grade range listed (6-8) for the program. Many problems involve a combination of math skills (not just +/-/÷/x). Preskill material should be provided with the program.

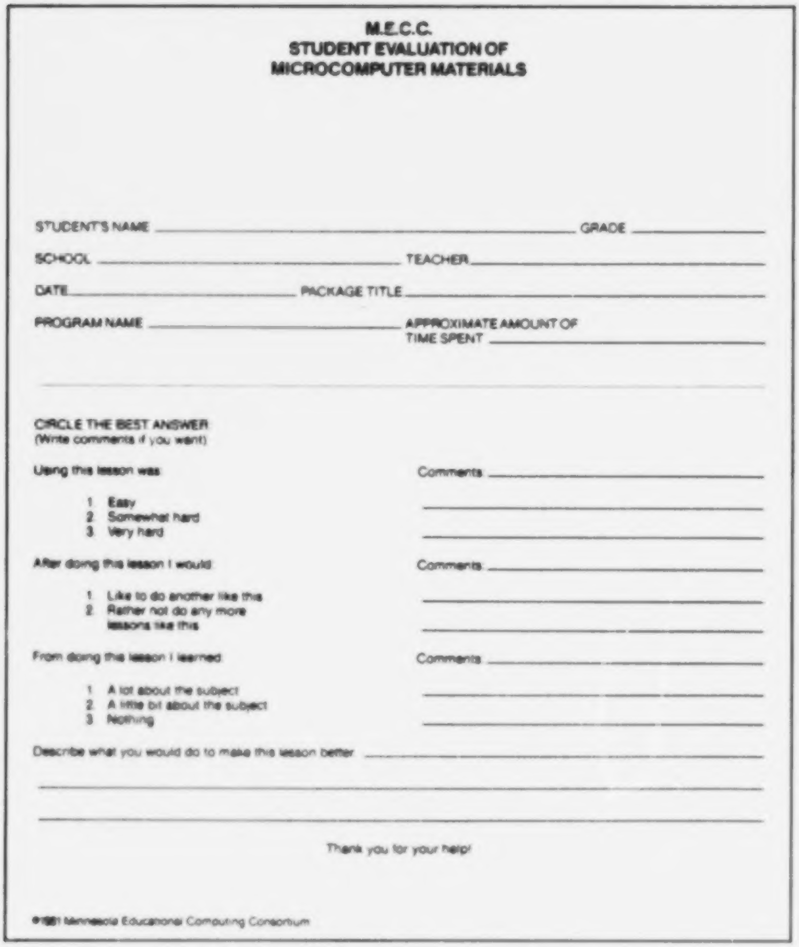
Other Comments:

Program examples are a little too fantastic for students to believe. Cute but highly unlikely.

Student Involvement. Observing students using a program can provide useful information about the strengths and weaknesses of the software. Students usually show a high degree of interest in reviewing software and can provide valuable comments about the program from a student perspective. Figure 5.4 shows an example of a form that students can use to evaluate microcomputer materials. Some guidelines for conducting student software reviews are listed below.

1. Select students who represent the student population for which a program is designed.
2. A teacher should assist students in using or developing a software evaluation form. The items on the form should be limited to those that students can easily comprehend and apply and should provide the adult(s) in charge with useful information that might be difficult to obtain without student input.
3. Assign students to work in pairs—one as a recorder, the other as the program operator. The roles of recorder and program operator should rotate between the students.
4. Limit the review period at the computer to forty minutes. An adult should observe the students at the computer, noting their reaction to the program and to the printed materials that are included with the program. Allow time to discuss the program and to complete the evaluation form with the students.

Evaluation Summary. During the final phase of the evaluation, the committee should discuss and reviews its findings. After differences have been resolved or clarified, a committee member should prepare a summary review. Reviews generally include objective and subjective details about the program and assign a value to the program using some type of scale. You can become an advocate for quality educational software by including evaluation comments with any program you return to a publisher. Your feedback may result in the program being improved or withdrawn from the market.



M.E.C.C.
STUDENT EVALUATION OF
MICROCOMPUTER MATERIALS

STUDENT'S NAME ____________________ GRADE __________
SCHOOL ____________________ TEACHER ____________________
DATE ____________________ PACKAGE TITLE ____________________
PROGRAM NAME ____________________ APPROXIMATE AMOUNT OF TIME SPENT ____________________

CIRCLE THE BEST ANSWER
(Write comments if you want)

Using this lesson was: Comments ____________________

1. Easy
2. Somewhat hard
3. Very hard

After doing this lesson I would: Comments ____________________

1. Like to do another like this
2. Rather not do any more lessons like this

From doing this lesson I learned: Comments ____________________

1. A lot about the subject
2. A little bit about the subject
3. Nothing

Describe what you would do to make this lesson better ____________________

Thank you for your help!

©1981 Minnesota Educational Computing Consortium

FIGURE 5.4

This has been taken from Student Evaluation Form, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.

The Review Process

If you do not have the time, resources, or desire to conduct a software evaluation, you can perform a software review, which is an abbreviated form of evaluation. A review usually involves one person guided by a short checklist of questions about a program's documentation, instructional design, and operation and does not include writing extensive comments.

EVALUATION QUESTIONS

Evaluating software is really a questioning process. How much does the program cost? Are backup copies provided? Is the program menu-driven? Does the program include support materials? What hardware is necessary to operate the program? Does the program present information in a way that is not easily duplicated with traditional classroom materials?

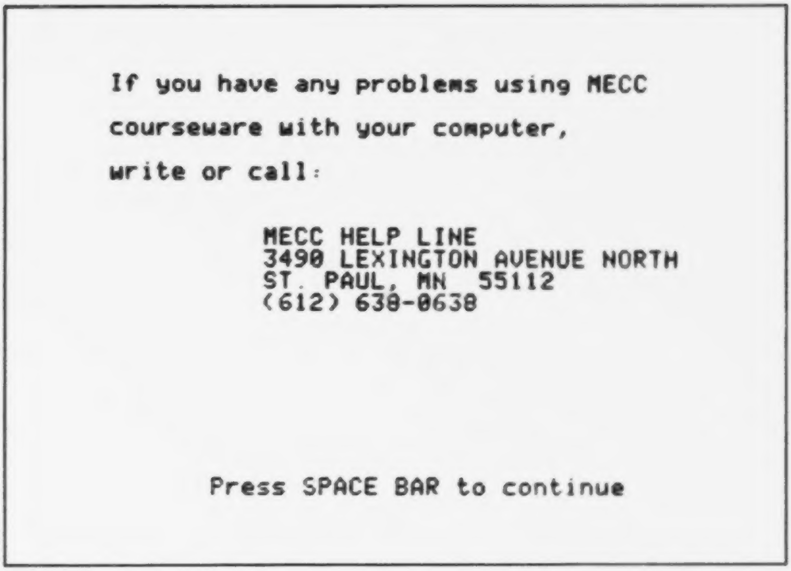
Hundreds of questions can be asked about a program. Numerous articles, books, evaluation guides, and checklists have attempted to present the important questions educators should ask about instructional software; however, no single evaluation method or list of questions is appropriate for every program. Each educator must identify and select the evaluation questions most relevant to his or her own needs.

As you examine the evaluation questions presented in the next section, try to establish your own mental checklist of questions you would want to ask about a program. The questions presented in this chapter will provide some examples for you to use. They apply to some, but not all, educational software; hence, you must exercise some judgment in applying them.

Preliminary Questions

There are many ways to categorize questions about educational software. Try to imagine the categories presented in this chapter as a series of tests that the software program must pass. If the answers to the questions in a category satisfy you, then proceed to the next set of questions about the program. If the answers are not satisfactory, you may decide to eliminate the program from your list of possibilities. The following preliminary questions are suggested.

- *Where can information about the program be obtained?* Methods for obtaining information about educational software were presented in the last chapter, but, in general, you can read about a program, ask about a program, or personally examine a program.
- *What equipment is needed to operate the program?* Before you examine a program, you must determine what type of microcomputer the program requires and if any peripheral devices are needed. Peripherals include additional disk drives, cassette tape drives, color monitors, printers, synthesizers, graphics tablets, light pens, joysticks, game paddles or mouses.
- *Will the program's publisher assist the consumer?* There are several ways that a software publisher can support a customer. For example, will the publisher provide the names and phone numbers of school districts where the program has been purchased? Will the publisher provide you with a preview copy of the program? Will the publisher provide special prices or reproduction rights for schools that buy multiple copies of a program? Is support service available from the supplier through a toll-free telephone number? The program, *Writing a Narrative,* provides the number for the publisher's "help line" on the screen (see Figure 5.5).
- *Does the company make adequate provisions for backup copies?* Programs sometimes become inoperable, are lost, or are stolen. If you plan to use a program and are dependent on its availability, you must have a backup copy. Backup policies vary from publisher to publisher, but some publishers provide a backup for a fee or after a damaged disk has been returned. Whenever possible, you should attempt to purchase a program that includes a backup disk. Figure 5.6 shows an example of Scholastic Software's warranty statement for the program *Microzine.*



```
If you have any problems using MECC

courseware with your computer,

write or call:

          MECC HELP LINE
          3490 LEXINGTON AVENUE NORTH
          ST. PAUL, MN  55112
          (612) 638-0638




     Press SPACE BAR to continue
```

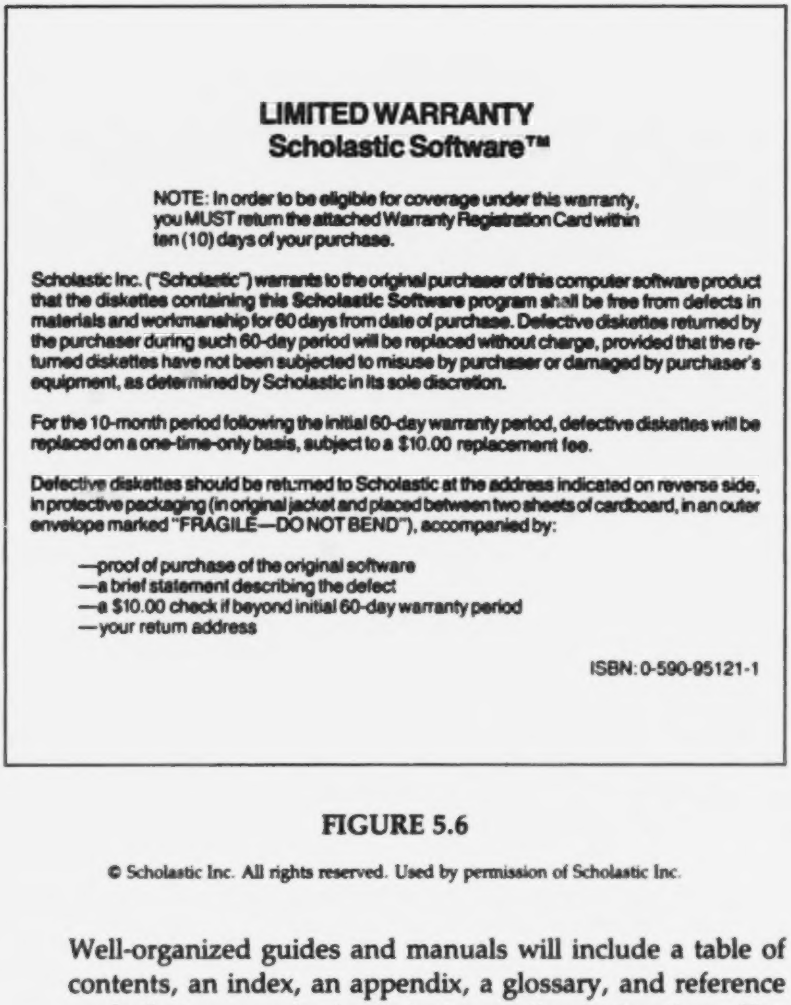
FIGURE 5.5

- *How much does the program cost?* Purchasing computer software is a financial investment. Most school budgets for software are quite limited; thus, you must determine your instructional objectives and establish the amount of money you are willing to spend to meet those objectives. It is also helpful to know what a program costs in comparison to similar programs.

Program Documentation

The next set of questions involves the printed material that accompanies the program, sometimes called documentation. The following questions should be asked about a program's documentation.

- *Is the documentation well-organized?* Poorly organized documentation is about as useful as no documentation at all.



LIMITED WARRANTY
Scholastic Software™

NOTE: In order to be eligible for coverage under this warranty, you MUST return the attached Warranty Registration Card within ten (10) days of your purchase.

Scholastic Inc. ("Scholastic") warrants to the original purchaser of this computer software product that the diskettes containing this **Scholastic Software** program shall be free from defects in materials and workmanship for 60 days from date of purchase. Defective diskettes returned by the purchaser during such 60-day period will be replaced without charge, provided that the returned diskettes have not been subjected to misuse by purchaser or damaged by purchaser's equipment, as determined by Scholastic in its sole discretion.

For the 10-month period following the initial 60-day warranty period, defective diskettes will be replaced on a one-time-only basis, subject to a $10.00 replacement fee.

Defective diskettes should be returned to Scholastic at the address indicated on reverse side, in protective packaging (in original jacket and placed between two sheets of cardboard, in an outer envelope marked "FRAGILE—DO NOT BEND"), accompanied by:

—proof of purchase of the original software
—a brief statement describing the defect
—a $10.00 check if beyond initial 60-day warranty period
—your return address

ISBN: 0-590-95121-1

FIGURE 5.6

© Scholastic Inc. All rights reserved. Used by permission of Scholastic Inc.

Well-organized guides and manuals will include a table of contents, an index, an appendix, a glossary, and reference cards that summarize important operating instructions. Figure 5.7 shows an example of a command summary card for the program *Delta Drawing*.

- *Does the documentation provide technical details?* A program's documentation should thoroughly specify what equipment is needed to run the program. Clearly written instructions about how to connect and use any peripherals that are required should also be presented.

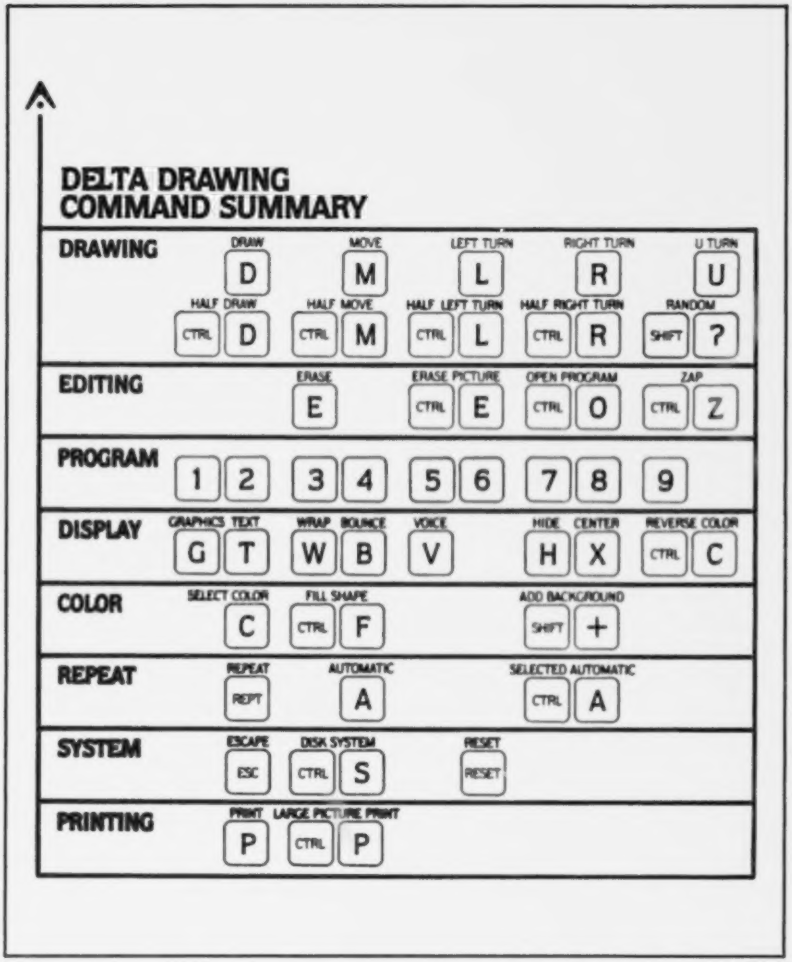

DELTA DRAWING
COMMAND SUMMARY
DRAWING
DRAW D
MOVE M
LEFT TURN L
RIGHT TURN R
U TURN U
HALF DRAW CTRL D
HALF MOVE CTRL M
HALF LEFT TURN CTRL L
HALF RIGHT TURN CTRL R
RANDOM SHIFT ?
EDITING
ERASE E
ERASE PICTURE CTRL E
OPEN PROGRAM CTRL O
ZAP CTRL Z
PROGRAM
1 2 3 4 5 6 7 8 9
DISPLAY
GRAPHICS G
TEXT T
WRAP W
BOUNCE B
VOICE V
HIDE H
CENTER X
REVERSE COLOR CTRL C
COLOR
SELECT COLOR C
FILL SHAPE CTRL F
ADD BACKGROUND SHIFT +
REPEAT
REPEAT REPT
AUTOMATIC A
SELECTED AUTOMATIC CTRL A
SYSTEM
ESCAPE ESC
DISK SYSTEM CTRL S
RESET RESET
PRINTING
PRINT P
LARGE PICTURE PRINT CTRL P

FIGURE 5.7

Reproduced with permission of Spinnaker Software.

- *Does the documentation provide a written tutorial?* Unless a program is extremely simple to use or is delivered to the user with a comprehensive demonstration, some kind of printed material will be needed to operate the program. Documentation that provides a written tutorial that teaches novices how to operate the program is essential. A well-written tutorial also includes printouts of screen displays from the program. Figure 5.8 shows an example of a tutorial page for the *Storytree* program.

- *Does the documentation address instructional concerns?* Documentation should include information about the program's mode of instruction (i.e., drill and practice, tutorial, simulation, or gaming), target audience, prerequisite skills, estimated student time needed for completion, and field testing.

  Does the documentation clearly define the program's instructional objectives? If so, you will want to know if the objectives are related to a specific instructional philosophy.
- *Does the documentation include instructional resources?* Some software documentation provides student worksheets related to the program along with useful resource lists for equipment, materials, and books. It is important to remember that support materials vary in quality, and worksheets will not compensate for poorly designed software.

An alternative for obtaining information about the content of a software program is to look at its menu (see Figure 5.9).

Screen Design

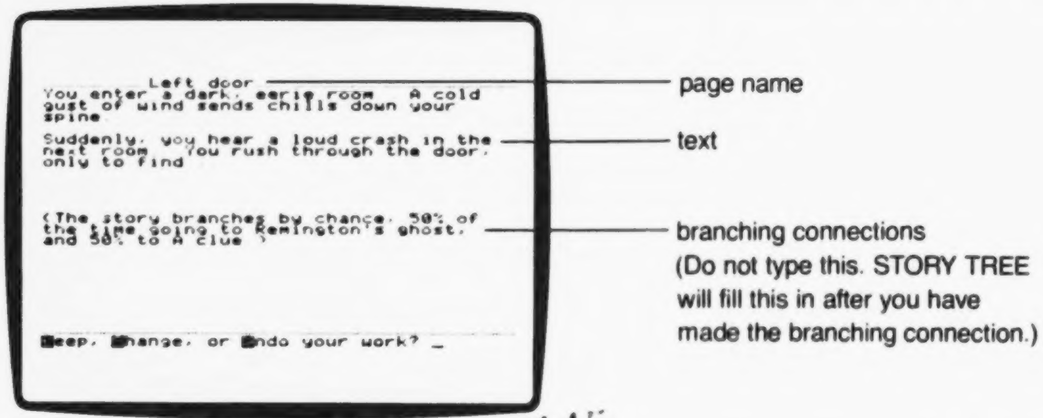
After evaluating a program's documentation, you should observe the program in action. This section suggests questions to ask when you watch a program run.

- *Is the screen well-organized?* Programs with well-organized screen displays avoid presenting too much text or graphics on the screen at one time and are divided into a number of functional areas that remain consistent throughout the course of the program. For example, the top part of the screen might be reserved for diagrams and the bottom part for explanation of those diagrams.
- *Who controls the screen?* The screen displays in some programs will advance automatically; however, all students do not read at the same speed. You should try to select pro-

FIGURE 5.8

Writing a CHANCE Page

You are now ready to work on the CHANCE page called 'Left door.' Here are the three parts of this page as they will look when you have entered them:



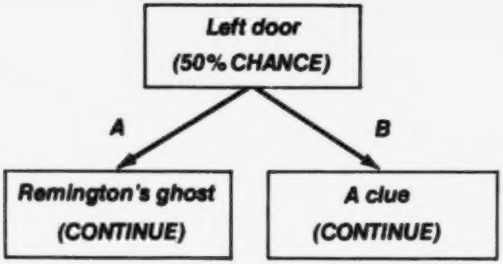
```
              Left door
You enter a dark, eerie room  A cold
gust of wind sends chills down your
spine

Suddenly, you hear a loud crash in the
next room  You rush through the door,
only to find


(The story branches by chance, 50% of
the time going to Remington's ghost,
and 50% to A clue )




Keep, Change, or Undo your work? _
```

page name

text

branching connections
(Do not type this. STORY TREE
will fill this in after you have
made the branching connection.)

BEST COPY AVAILABLE


Left door
(50% CHANCE)
A
B
Remington's ghost
(CONTINUE)
A clue
(CONTINUE)

To write this page:

1 Select 'Left door' to work on.
2 Press RETURN to accept the page name.
3 Type the page text and press RETURN.

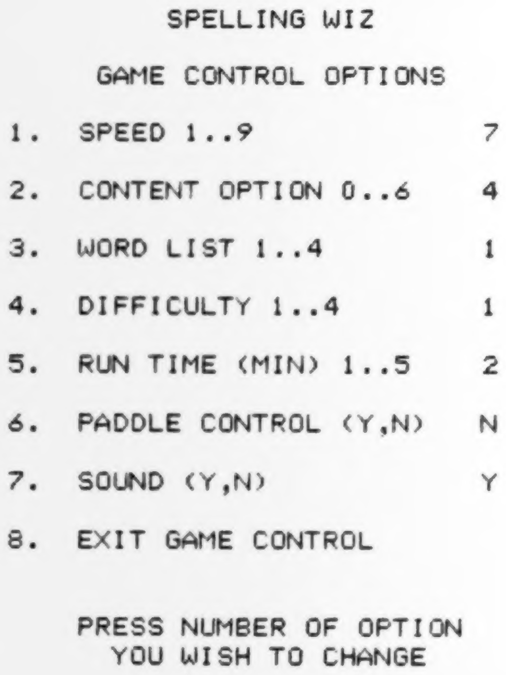
© Scholastic Inc. All rights reserved. Used by permission of Scholastic, Inc.

BEST COPY AVAILABLE

Examining a Program's Menu

There are several ways to obtain a quick overview of a program's content. One way is to read the documentation that accompanies a program. Another approach is to load the program and study its menu(s).

A menu is a screen display designed to present the user with a number of fixed options that can be selected. Menus provide a means for the user to branch to various parts of a program; if properly designed, they can provide a significant amount of orientation information. A program's menu can tell you how many options the program includes and who controls the options—the student or the teacher. For example, by studying the *Spelling Wiz* menu below, you can obtain a sense of how the program is organized.



```
              SPELLING WIZ

          GAME CONTROL OPTIONS

1.   SPEED 1..9                    7

2.   CONTENT OPTION 0..6           4

3.   WORD LIST 1..4                1

4.   DIFFICULTY 1..4               1

5.   RUN TIME (MIN) 1..5           2

6.   PADDLE CONTROL (Y,N)          N

7.   SOUND (Y,N)                   Y

8.   EXIT GAME CONTROL

      PRESS NUMBER OF OPTION
        YOU WISH TO CHANGE
```

FIGURE 5.9

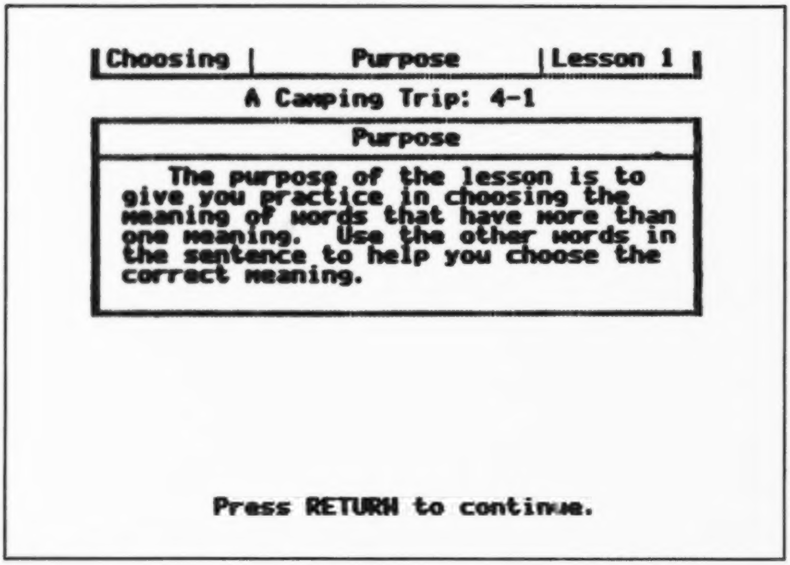
© DLM Teaching Resources, Allen, TX 75002.

grams that allow the user to control the timing of screen advancement. Figure 5.10 shows a screen display from *Custom Ware*. The screen is organized into functional areas, and the user controls the screen.

- *Does the program have a consistent display rate?* Some programs are written in programming languages that process information slowly, and images that appear on a screen at a slow or inconsistent speed can be distracting to students. Consider the pacing of a program as you would consider the pacing of instruction presented by a teacher. Well-written programs allow the teacher or student to control presentation rates.

Program Instructions

- *How does the program present instructions?* One of the first questions you will ask when you examine a program is



```
Choosing |      Purpose      | Lesson 1
       A Camping Trip: 4-1
              Purpose
  The purpose of the lesson is to
give you practice in choosing the
meaning of words that have more than
one meaning.  Use the other words in
the sentence to help you choose the
correct meaning.

      Press RETURN to continue.
```

FIGURE 5.10

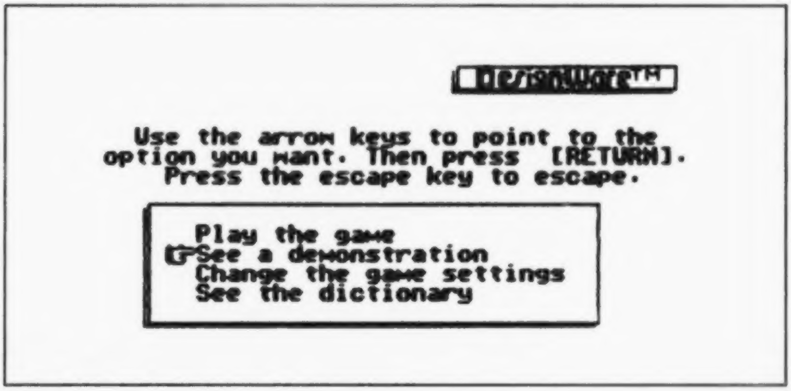
BEST COPY AVAILABLE

"How do I make it work?" Do you need to read the program's manual first or can you operate the program without it? A well-organized program allows the user to skip or review instructions at any point in the program.

- *Are the instructions easy to understand?* You must also ask yourself how your students will learn to operate the program. Will you need to explain the program in detail to each student, or will the students be able to operate the program independently? You must also decide if the instructions are confusing or beyond the reading level of your students. Some programs, like *Trap-a-Zoid,* include a demonstration of how the program operates (see Figure 5.11).

Program Interaction

- *Does the program require a lot of typing?* If your students are not accomplished typists, you should try to select programs that do not require a lot of interaction with the keyboard. Some programs permit the use of entry devices such as joysticks, game paddles, or a light pen. Alternative entry



Use the arrow keys to point to the option you want. Then press [RETURN]. Press the escape key to escape.

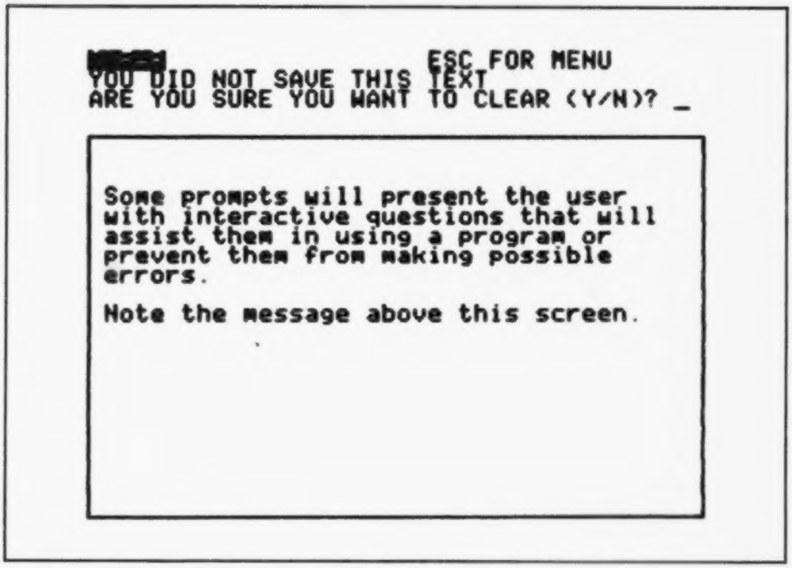
FIGURE 5.11

Screen from Trap-a-Zoid,™ a geometry program by Designware®
Britannica Learning Corp., San Francisco, CA

devices are especially important if a program is to be used with handicapped students or very young children. A program should not penalize students for poor typing skills--unless the objective of the program is to improve their typing skills.

- *Do cues and prompts help students operate the program correctly?* A number of prompts or cues can be used to assist students in a program. For example, symbols may be used to indicate that the student is to perform a specific action such as pressing the RETURN key. A program may also provide a student with a hint or clue before providing the correct answer. Figure 5.12 shows how *The Bank Street Writer* word-processing program prompts the user to consider carefully whether or not he or she wishes to clear the screen.

A common prompt used in many CAI programs is blinking and flashing. Blinking and flashing of text should be



```
                   ESC FOR MENU
YOU DID NOT SAVE THIS TEXT
ARE YOU SURE YOU WANT TO CLEAR (Y/N)? _
```

```
Some prompts will present the user
with interactive questions that will
assist them in using a program or
prevent them from making possible
errors.

Note the message above this screen.
```

FIGURE 5.12

© Scholastic Inc. All rights reserved. Used by permission of Scholastic Inc.

used sparingly in a program, and it should never blink or flash two separate areas of the screen at the same time competing for the student's attention.

- *Does the program encourage or alienate the student?* A program should never belittle a student with negative statements, sounds, or graphics when the student makes an error. The statement, TRY AGAIN, can also be punishing if the program allows the student to repeat an error (see Figure 5.13). A well-designed program will provide the student with help and allow him or her to repeat the incorrect item. Some programs periodically review items that are incorrect.

  Some programs acknowledge a correct item with superlatives such as MAY THE FORCE BE WITH YOU, YOUNG GENIUS! Students sometimes respond to such programs with disdain. It is also a good idea to select a program that presents intermittent and varied reinforcement because the same reward message given over and over again may eventually become meaningless to the student.

Error Correction

Some instructional progams permit students to make several errors before providing a correction. As a result, students may actually be practicing incorrect answers, thus making subsequent remediation more difficult. A well-designed program will either correct a student's error immediately, followed by a review of the incorrect item, or cycle the student out of the program with a message instructing the student to request assistance from his or her teacher. For example, the following dialogue between a student and computer illustrates appropriate error correction.

```
C:  You seem to be having difficulty with the meaning of
    pre.

S:  Pre means before.
    Pre history means before history.
    Pre wrap means to wrap before.

    What does pre mean?
```

S: Before

C: Good! So what word means to write before?

S: Prewrite

C: Excellent! Let's practice several more examples.

Some programs use the ad nauseum approach to error correction. For example:

C: 4 × 6 = ?

S: 12

C: Wrong answer. Try again. 4 × 6 = ?

S: 25

C: Wrong answer. Try again. 4 × 6 = ?

S: 21

C: Wrong answer. Try again. 4 × 6 = ?

S: I hate you!

This type of error correction procedure continues until the student stumbles across the correct answer. In this instance, the infinite patience of the computer becomes an irritating attribute.

A variation of the ad nauseum approach is to provide the student with the correct answer after three consecutive errors. The flaw in this strategy is the lack of cumulative review of problems that a student answers incorrectly. A cumulative review of errors keeps difficult items "alive" for the student and is necessary in order to determine if the student can provide a correct answer on the first trial.

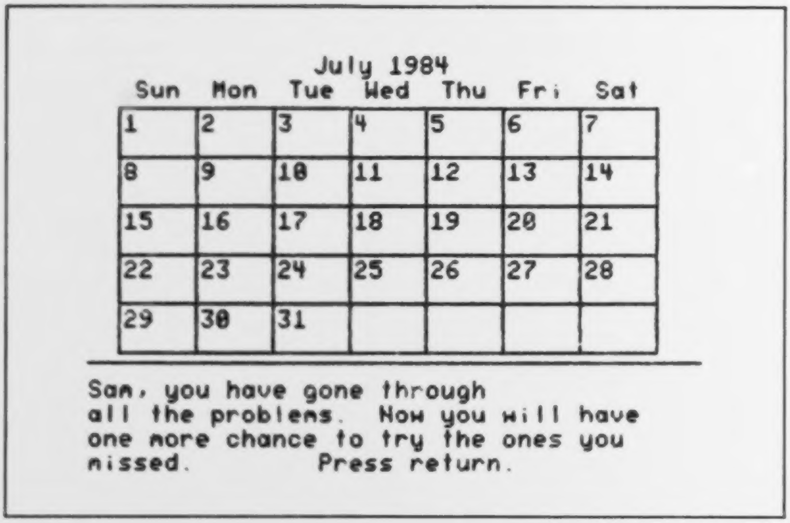
FIGURE 5.13

- *Does the program allow for a wide range of student responses?* Students will respond to questions in CAI programs in various ways. For example, a program might display a map of Tennessee and ask the student to type the name of the state. How should the program respond if the student types Tennasee? In general, a program should not penalize a student for misspelling a word in a lesson that does not have spelling as an objective. It should be designed to recognize a range of correct and incorrect answers.

  A carefully written program will accommodate all reasonable typing errors without disruption. You should test a program's error handling by deliberately making mistakes. For example, if a program asks you to respond by typing the letters Y or N, type a different letter and observe the results. A well-designed program will either correct the error or respond with a follow-up question, while a mediocre program will "crash," or cease to operate.

Instructional Content

- *Does the program present content in a logical order?* Care must be taken not to overload the short-term memory of students. A program should introduce new information or concepts in small units and provide the student with opportunities for immediate practice. A well-designed program should also incorporate a review of important concepts or difficult items. Figure 5.14 shows a screen display from *Using a Calendar.* The message on the screen display indicates that the program recycles items missed by the student.
- *Is the program's content transferable and generalizable?* Skills taught via a computer program should be transferable to other environments. For example, a student may learn from the computer how to construct an electrical circuit, but not be able to complete the task when it is attempted with a battery, wires, and light bulb. Skills that can be applied only with the computer are of limited value.



```
                July 1984
 Sun   Mon   Tue   Wed   Thu   Fri   Sat
 1     2     3     4     5     6     7
 8     9     10    11    12    13    14
 15    16    17    18    19    20    21
 22    23    24    25    26    27    28
 29    30    31

Sam, you have gone through
all the problems.  Now you will have
one more chance to try the ones you
missed.         Press return.
```

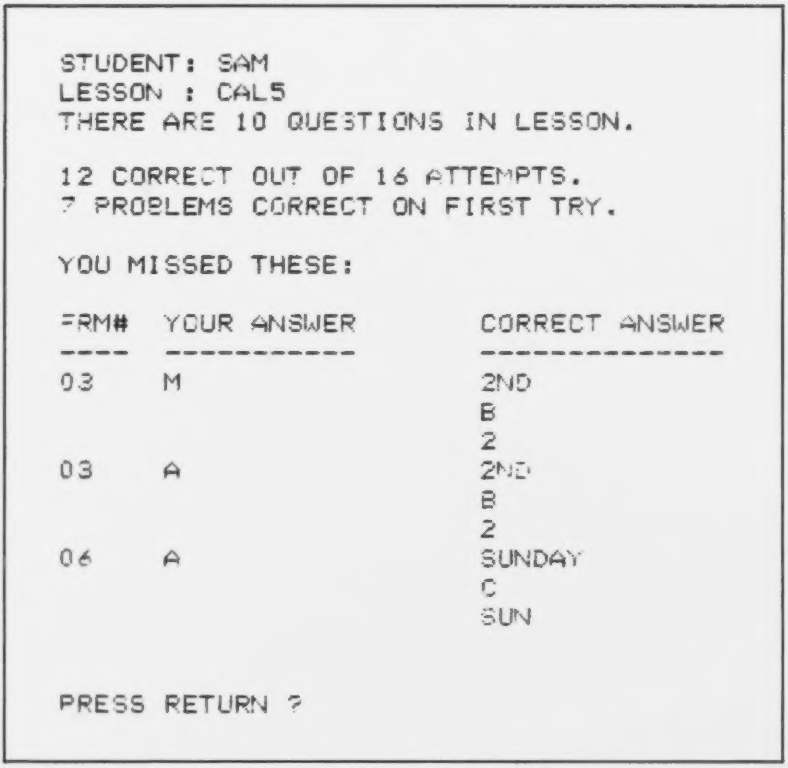
FIGURE 5.14

- *Is the content consistent with the program's objectives?* Some programs do not deliver the results promised. The best way to determine how well a program meets its objectives is to examine carefully the behavior that the program expects from the students.
- *Is the program's vocabulary level appropriate for the target audience?* CAI programs frequently require a lot of reading. Like textbook materials, some programs use vocabulary that is beyond the reading level of the intended audience.
- *Is the program's content accurate, unbiased, and current?* Computer programs should present information that is factually correct and up to date. The content should be unbiased and free from stereotypes regarding religion, ethnic background, race, sex, and age.

Record Keeping and Management

- *How does the program record student responses?* Most CAI programs have a built-in score keeping system. Some systems simply record the number of correct and incorrect responses, while a more sophisticated CAI program will report exactly which items a student is having diffficulty with, how many items were answered, and how long the student took to complete the program. Figure 5.15 shows a screen display from *Using a Calendar*. Note the various records that the program presents.

- *Does the program provide statistical information on a student's progress?* Computers are the ultimate number-crunching machines, and it's reasonable to expect that a program will provide more than just a student's raw score. Many pro-

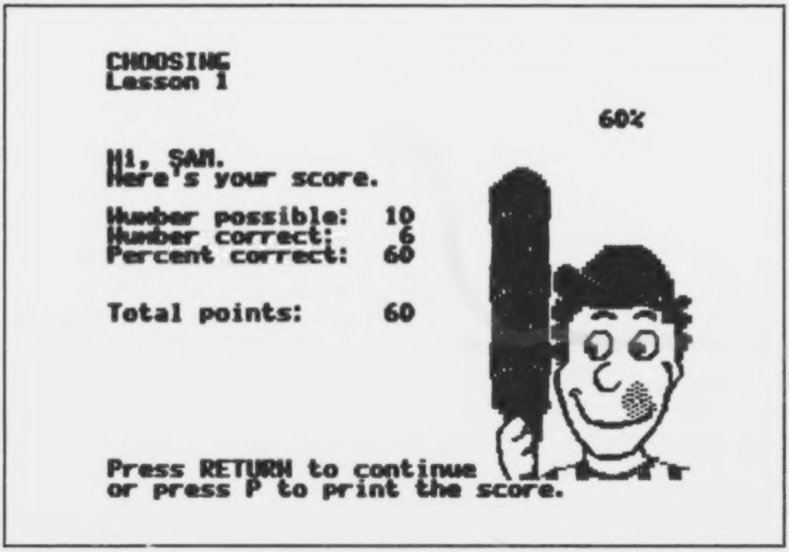


```
STUDENT: SAM
LESSON : CAL5
THERE ARE 10 QUESTIONS IN LESSON.

12 CORRECT OUT OF 16 ATTEMPTS.
7 PROBLEMS CORRECT ON FIRST TRY.

YOU MISSED THESE:

FRM#  YOUR ANSWER        CORRECT ANSWER
----  -----------        --------------
03    M                  2ND
                         B
                         2
03    A                  2ND
                         B
                         2
06    A                  SUNDAY
                         C
                         SUN

PRESS RETURN ?
```

FIGURE 5.15

grams will calculate a student's percentage or average score. Some programs will graphically illustrate student or class data using histograms, pie charts, or line graphs. Figure 5.16 shows an example of how the *Custom Ware* program graphically represents a percentage score using scoops on an ice cream cone.

- *Does the program display and maintain records?* Record-keeping data should be available on the screen and via a printout. Printouts of scores can be examined by the teacher at a later date or presented to students and their parents for evaluation purposes. Some programs present data that vanishes when the student presses RETURN. This record-keeping design is useless to the teacher unless he or she is constantly watching the student's screen. The burden for keeping records should be placed on the computer and not on the teacher. A well-designed program automatically records student scores, updates files, and allows information to be



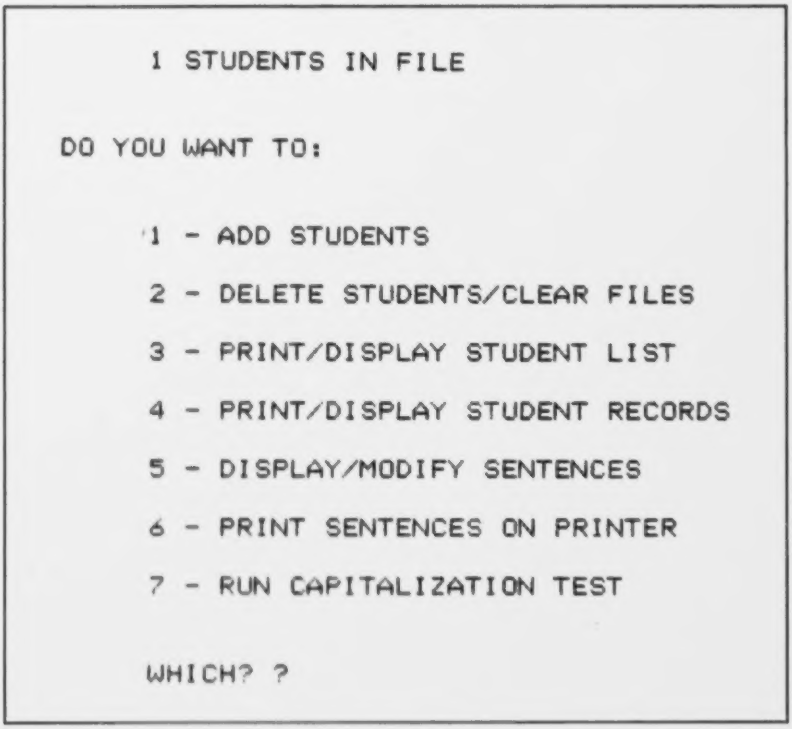
```
CHOOSING
Lesson 1
                                     60%

Hi, SAM.
Here's your score.

Number possible:   10
Number correct:     6
Percent correct:   60

Total points:      60



Press RETURN to continue
or press P to print the score.
```

FIGURE 5.16

retrieved or deleted easily. Figure 5.17 shows an example of the menu from *Capitalization*. The menu indicates that the program permits student records to be saved on a disk, printed, or displayed on the screen.

Final Questions

- *Is the program interesting to students?* A program may appear to be fun and interesting to you as an adult, but you should reserve final judgment until you observe the program being used with students. A program may have all the desirable features on paper and still be boring to students. In other words, it is wise to involve students in the evaluation and review process whenever possible.



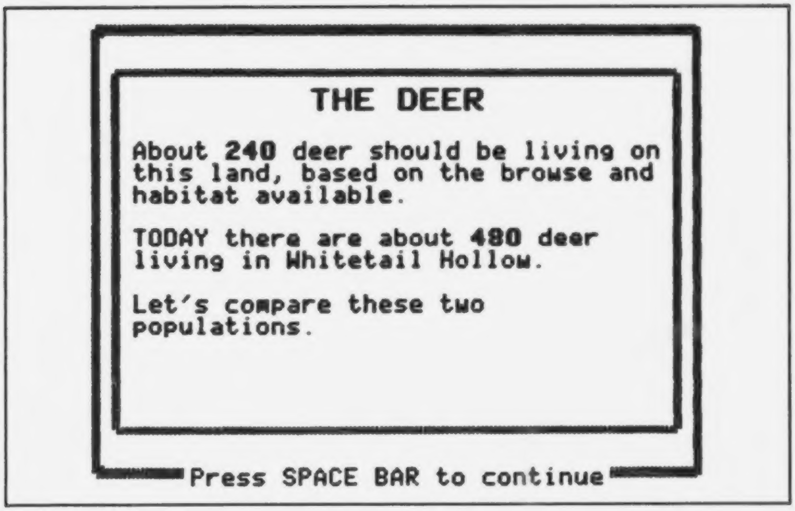
```
          1 STUDENTS IN FILE

     DO YOU WANT TO:

          1 - ADD STUDENTS

          2 - DELETE STUDENTS/CLEAR FILES

          3 - PRINT/DISPLAY STUDENT LIST

          4 - PRINT/DISPLAY STUDENT RECORDS

          5 - DISPLAY/MODIFY SENTENCES

          6 - PRINT SENTENCES ON PRINTER

          7 - RUN CAPITALIZATION TEST

          WHICH? ?
```

FIGURE 5.17

- *Is the program effective compared to noncomputer instruction in the same area?* After you have carefully examined a program, you should try to determine if you can accomplish the same objectives as well (or better) by using noncomputer materials. In the final analysis, you must decide whether or not a computer is the best way to accomplish your instructional objectives.
- *Does the program take advantage of the computer's special capabilities?* An important part of evaluating educational software is assessing how effectively a program incorporates the computer's capabilities. Quality instructional sequences require two-way communication. A student should not be a passive viewer of information presented on the screen. Programs that use the computer as an automatic page turner are poor substitutes for books.

  The use of graphics and sound are often mentioned in regard to the computer's ability to integrate various forms of communication. Sound and graphics can be used to enhance the presentation of ideas and the aesthetics of a program. Figures 5.18 and 5.19 show how the *Oh, Deer!* program uses graphics to illustrate the concept of population density.

  The use of sound and graphics in a program can also be distracting; for instance, a long, loud, attention-getting sound in a program can disrupt the entire class. A well-designed program will provide the option to turn off the sound when desired.
- *Can the program content or operation by easily modified?* A well-designed program will permit you to delete or add items easily. In effect, such programs allow teachers to customize the program for their students. It is also helpful to be able to adjust the way a program operates. For example, you may wish to have items presented to students in random order, or you may want to provide the student with assistance after a certain number of errors. Figure 5.20 shows a screen display from *Add-Your-Own-Content,* a program that allows teachers to specify practice items.



THE DEER

About 240 deer should be living on this land, based on the browse and habitat available.

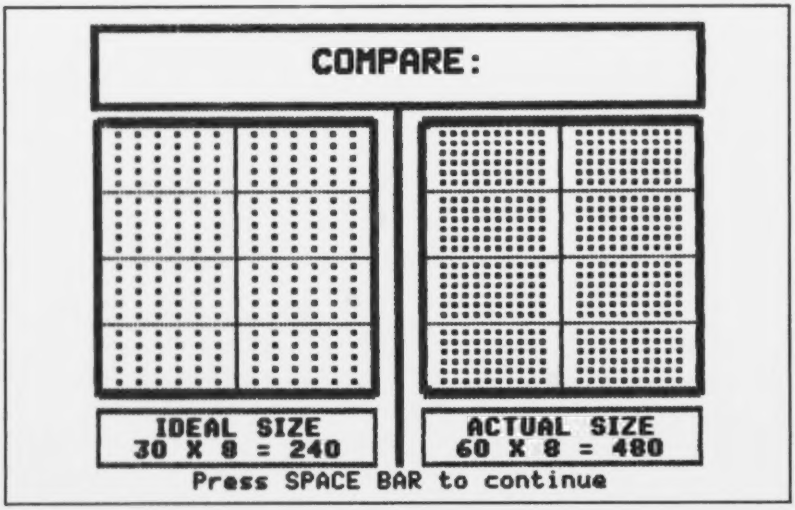
TODAY there are about 480 deer living in Whitetail Hollow.

Let's compare these two populations.

Press SPACE BAR to continue

FIGURE 5.18

This has been taken from *Oh, Deer!*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.



COMPARE:

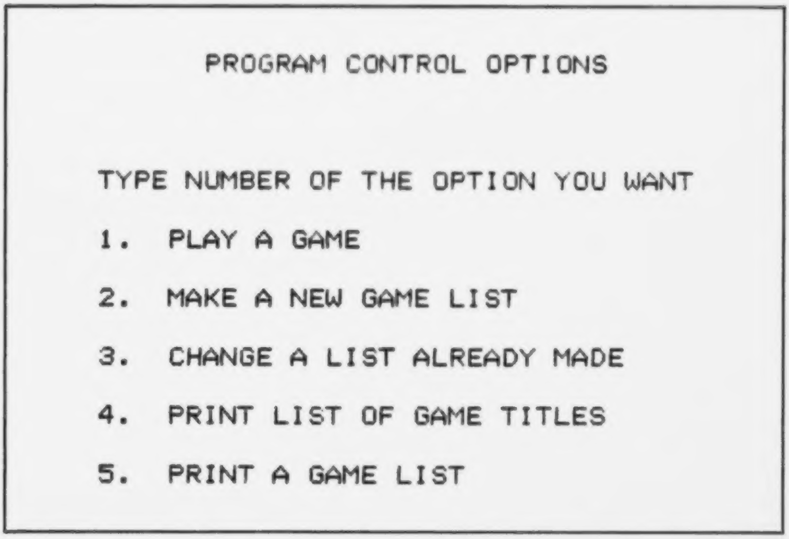
IDEAL SIZE
30 X 8 = 240

ACTUAL SIZE
60 X 8 = 480

Press SPACE BAR to continue

FIGURE 5.19

This has been taken from *Oh, Deer!*, published by MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.



```
PROGRAM CONTROL OPTIONS


TYPE NUMBER OF THE OPTION YOU WANT

1.  PLAY A GAME

2.  MAKE A NEW GAME LIST

3.  CHANGE A LIST ALREADY MADE

4.  PRINT LIST OF GAME TITLES

5.  PRINT A GAME LIST
```

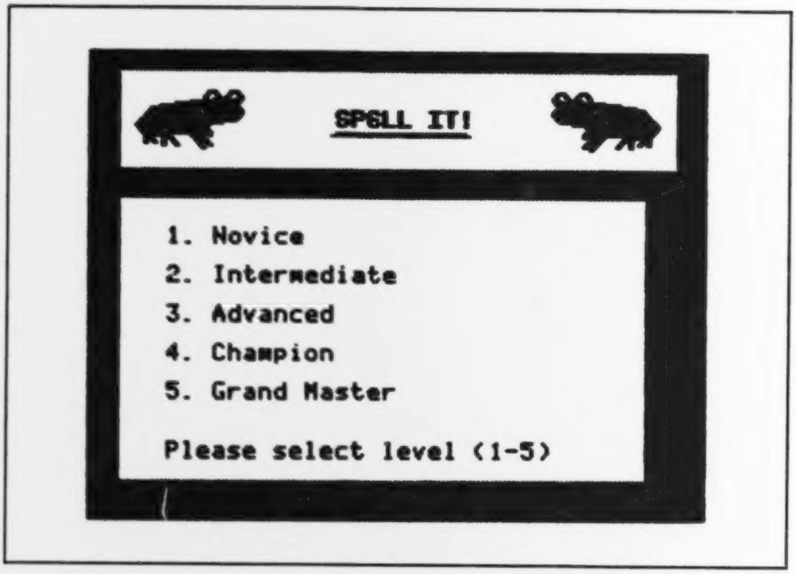
FIGURE 5.20

© DLM Teaching Resources, Allen TX 75002.

In some cases, it is desirable to allow the student to determine what content or skill level should be presented. Figure 5.21 shows a screen display from *Spell It!* The display indicates that the learner can select his or her own skill level to practice spelling. Figure 5.22 shows a screen display from *Using a Calendar*. The message on the display asks the student to identify the lesson he or she wishes to work on.

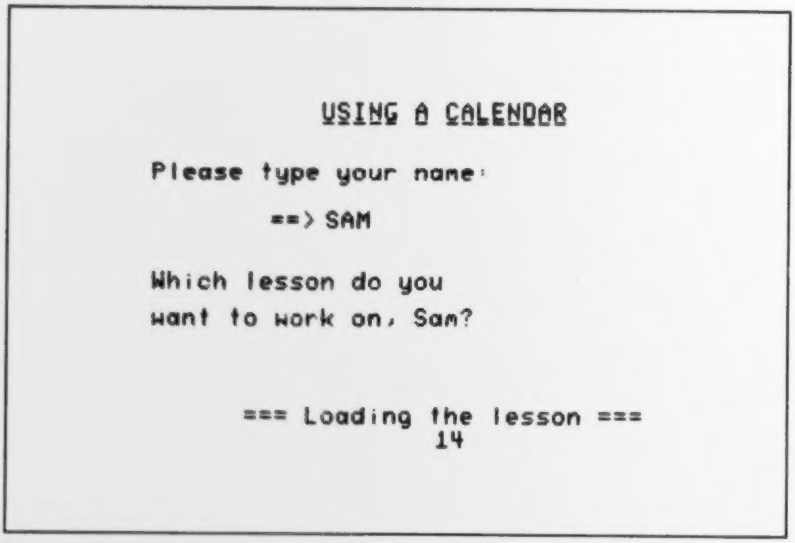
CONCLUSION

Whether you're conducting a software evaluation or a software review, remember that evaluative criteria in the area of computer software are both numerous and tentative. No single program can incorporate every attribute of a well-designed program. Software selection requires compromises based on thorough knowledge of



```
SPELL IT!

1. Novice
2. Intermediate
3. Advanced
4. Champion
5. Grand Master

Please select level (1-5)
```

FIGURE 5.21



```
            USING A CALENDAR

Please type your name:

        ==> SAM

Which lesson do you
want to work on, Sam?


      === Loading the lesson ===
                   14
```

FIGURE 5.22

the field; however, this does not mean that you must compromise yourself to the point of accepting mediocre programs.

Software evaluation forms, critera, and procedures continue to be refined. A single, general-purpose software evaluation form may have limited practical value for two reasons: (1) in trying to evaluate everything, the instrument does nothing really well, and (2) effective use of an instructional program requires a match with the local curriculum—and a software evaluation instrument designed to cross school boundaries cannot do this very well. One solution is to stress the use of separate evaluation instruments for different products and to emphasize the limitations of any procedure that does not focus on the integration of a product into the local curriculum.

For the present, you may find that it is more appropriate to develop your own evaluation criteria based on your students' needs and skills and the type of program you are evaluating. A list of thoughtful evaluation questions can direct your focus toward questions that you might not otherwise consider. Finally, remember that the acid test of a program is to field-test it with students.

READINGS AND RESOURCES

Evaluation of Educational Software: Guide to Guides. Northeast Regional Exchange, Southwest Educational Development Laboratory, 211 East 7th Street, Austin, TX 78701, 1983.

This book describes different software evaluation models and systems and contains a number of examples of evaluation instruments and software reviews.

Lathrop, Ann, and Goodson, Bobby. *Courseware in the Classroom.* Menlo Park, CA: Addison-Wesley Publishing Company, 1983.

This book is an excellent broad-based reference on all aspects of selecting, organizing, and using educational software. It is clearly written and reflects the wisdom of two widely respected practitioners of computer education. It is a useful book for any computer-using educator.

The following products have been included as examples of educational software in this chapter.

1. *Writing a Narrative* (Apple II+/IIe)
   Oh, Deer! (Apple II+/IIe)
   Publisher: Minnesota Educational Computing Consortium
   3490 Lexington Avenue North
   St. Paul, MN 55112

2. *Custom Ware: Multiple Meaning Words* (Apple II+/IIe)
   Publisher: Random House, Inc.
   201 East 50th St.
   New York, NY 10022

3. *Trap-a-Zoid* (Apple II+/IIe)
   Publisher: Designware, Inc.
   185 Berry St.
   San Francisco, CA 94107

4. *Using a Calendar* (Apple II+/IIe)
   Nouns and Pronouns (Apple II+/IIe)
   Capitalization Apple II+/IIe)
   Publisher: Hartley Courseware, Inc.
   Box 431
   Diamondale, MI 48821

5. *Spell It!* Apple II+IIe)
   Publisher: Davidson and Associates
   6069 Groveoak Place #12
   Rancho Palos Verdes, CA 90274

6. *Add-Your-Own-Content* (Apple II+/IIe)
   Publisher: Developmental Learning Materials
   One DLM Park
   Allen, TX 75002
   Available For: Apple II

7. *The Bank Street Writer* (Apple II+/IIe, Commodore 64, IBM PC, Atari 400/800)
   Publisher: Scholastic, Inc.
   P.O. Box 7502
   Jefferson City, MO 65102

Sources of Software Evaluation Forms

Guidelines For Evaluating Computerized Instructional Materials

The evaluation criteria in this document are comprehensive and easy to use. The checklist requires only making checks, circling words, or bracketing ranges of dots on judgment scales. Published by:

National Council of Teachers of Mathematics
1906 Association Drive
Reston, VA 22091

The Evaluator's Guide For Microcomputer-Based Instructional Packages

This guide was originally developed to be used by school personnel participating in the software evaluation process of MicroSIFT. The evaluation criteria used in this booklet are clearly defined and quite extensive. Published by:

The Computing Teacher
University of Oregon
1787 Agate St.
Eugene, OR 97403–1923

Microcomputer Courseware Evaluation Form

The organization of this form is appropriate for a systematic software evaluation. The form requires the user to write extensive comments and descriptions. Published by:

EPIE & Consumers Union
P.O. Box 620
Stony Brook, NY 11790

Computer Courseware Review Forms
Student Evaluation of Microcomputer Materials
The Microcomputer Educational Materials Evaluation

The MECC evaluation forms are appropriate for discussion and comparison of software by a committee. They are quite easy to use and are especially recommended for the novice. Published by:

Minnesota Educational Computing Corporation
2520 Broadway Drive
St. Paul, MN 55113–5199

REFERENCES

Evaluator's Guide for Microcomputer-Based Instructional Packages. Eugene, OR: International Council for Computers in Education, 1983.

Gleason, G. T. "Microcomputers in Education: The State of the Art." *Educational Technology* 21 (1981):7–18.

PROJECTS AND QUESTIONS

1. Evaluate an instructional computer program using one of the evaluation guides listed in the References and Resources section of this chapter.
2. Create a list of ten questions that should be asked about any educational software program. Briefly summarize the rationale for each question specified in the list.
3. Identify the major differences between a software evaluation and a software review.
4. React to the following statements in writing or in a class discussion.
   a. The most important software evaluation questions relate to how easily a program can be used.
   b. Generally speaking, software evaluation procedures are not well developed.
   c. Judging a software product fairly includes comparison among competing products.
5. Create a list of evaluation questions for a specific category of educational software. For example, create a list of questions that might be used to evaluate instructional games. Use your list of questions to evaluate a program from the software category you select.
6. Select five educational programs at random. Try to use each program without looking at the program's documentation. Describe the results.

7. Work in a group of five people to accomplish the following tasks:
    a. Select an evaluation form and one program to evaluate.
    b. Each member of the groups should evaluate the program individually using separate evaluation forms.
    c. When each member of the group has evaluated the same program, the group should meet and discuss their individual evaluations.
    d. Summarize the results in writing.
8. Organize a group of students to use a program and evaluate it using the questions listed on the MECC student evaluation form shown in this chapter. Ask the students to discuss their evaluations with each other. Summarize the results in writing.

6

Organizing and Using Educational Software

INTRODUCTION

Reading and thinking about the past five chapters represents a considerable investment in learning about issues related to selecting educational software, but you must also learn how to organize and use software intelligently. You must consider such questions as where the programs will be located, who will be responsible for them, and what the best way is to use a particular program. This chapter focuses on how to organize and use software in a school setting.

ORGANIZING A SOFTWARE COLLECTION

The location of software and decisions about its use can affect how it is used by a school staff. If, for example, software is not kept in a central location, its use will be limited. Possible locations for housing a software collection will vary from school to school; however, some common factors that will influence decisions about placement include the physical layout of the school, security concerns, the amount and location of computer equipment, and the needs of the staff and students.

Policies and Procedures

Decisions about how to locate and organize a software collection should be based on input from users. Staff members, students, and parents should be consulted. While ease of access should be one goal in deciding where to locate a software collection, the protection and preservation of the materials for future use should also be considered. Computer software is not immortal! According to the American Association of School Librarians (1983), "the life of a floppy disc appears to be around forty access hours."

Lathrop and Goodson (1983) recommend the development of a policies and procedures manual to help ensure that software is protected but accessible. Such a manual, when developed by a school or a school district committee, can reduce potential conflicts by providing both the staff and students with information related to the software collection. The issues to be addressed in a policies and procedures manual include:

1. Who is responsible for selecting, evaluating, ordering, cataloging, storing, and monitoring the software.
2. What procedures will be used to select, evaluate, order, catalog, store, and monitor the software.
3. How the software will be protected from illegal copying.
4. Where the software collection will be located.
5. How users will locate specific software titles.
6. How the software will be checked out.
7. Where the software can be used.
8. Who will have access to the software collection.
9. How long a program can be checked out.
10. Whether or not fines will be imposed for overdue or damaged software.
11. How complaints about policies, procedures, or software will be dealt with.
12. Who will approve the policies and procedures.

Once policies and procedures are agreed upon, they must be publicized. A short staff meeting, flyers, or posters can be used to inform the staff and students about procedures and policies that have been approved. Finally, the policies and procedures manual should be reviewed periodically to determine if changes need to be made. Many problems can be avoided by careful planning, diplomacy, and common sense.

Cataloging Software

Printed catalog cards are usually not supplied by software publishers. When software arrives at a school or a school district office, a complete set of catalog cards should be prepared and filed in the main catalog in the library media center. You may also find it advantageous to prepare a duplicate set of cards for use in a computer lab or classroom.

At the present time there is no standard procedure for cataloging microcomputer software; however, the Dewey Decimal Classification system and the *Sears List of Subject Headings* seems to be the most popular system used to classify and catalog software. This system is familiar to teachers and students, and it allows catalog cards for software to be integrated with the existing catalog system for books and other media.

Figure 6.1 shows a sample catalog card that can be modified to suit local practices. A program can be cataloged by subject, title, or author by following the steps specified below.

1. Assign the software a call number from the Dewey Decimal Index according to the subject matter. For example, the science call number would be 510. The call number is entered in the upper left corner of the card catalog.
2. Select one of the following descriptors to indicate the type of media and place it above the call number: Courseware, Microcomputer Program (or MCP), Computer Program (or CP). The term "machine-readable data file" is sometimes used as a descriptor for software; however, the term is meaningless to many students and teachers. A common

A Sample Main Entry Catalog Card

MICROCOMPUTER PROGRAM
510
SE
The Secrets Of Linear Algebra
One 5¼" diskette, Guide 76 p.
Synopsis: Designed to introduce students to systems of linear equations and their solution by means of elementary row operations on matrices.
Equipment: IBM PCjr microcomputer and monitor.
1 Mathematics-Algebra

FIGURE 6.1

practice used in library media centers to indicate audiovisual materials is to place a colored band across the top of the catalog card.

3. Place the first two letters of the first word in the title below the call number. Indefinite and definite articles at the beginning of a title (A, An, The, etc.) should be ignored. For example, the call number for a math program titled The Secrets of Linear Algebra would appear as:

    MICROCOMPUTER PROGRAM
    510
    SE

4. Use the *Sears List of Subject Headings* to determine which subject(s) the program most represents. If the Sears list does not have a relevant heading, create your own heading and record it in the office copy of the Sears list. Subheadings such as Computer Tutorial or Computer Game can be added to indicate the type of program.
5. A catalog card should specify bibliographic information about a program such as the title, producer or publisher, and copyright date.

6. A physical description of the software should cite all information about it. For example:
    - Floppy disk, hard disk, cartridge, or tape.
    - An itemized documentation list.
    - Brand and model of microcomputer needed for use (Commodore 64, TRS-80, etc.).
    - Memory requirements (16K, 32K, etc.).
    - Operating system (DOS 3.3, CP/M, etc.).
    - List of compatible equipment and peripherals necessary for the program to run (printer, graphics tablet, etc.).
7. A catalog card should also provide a brief synopsis of what the program offers.

A common problem in cataloging software is the inclusion of more than one program on a single disk. One solution is to create a title analytic card that identifies the individual programs. Another solution is to use a database program to file a variety of information about programs in a software collection. A database catalog of a software collection would permit a user to retrieve the names of specific programs rapidly on the basis of title, grade level, subject matter, and so on.

Processing Software

Once software has been cataloged, it must be prepared for storage and use. Some important factors to consider when processing software are discussed below.

Storage. An important first step in processing software is to decide how the software collection will be stored—that is, in hanging racks, on shelves, or in a file cabinet. The storage location you select should keep the software collection safe from theft yet accessible to whomever is permitted to use the software. Disks should be stored vertically in a container and protected from extreme temperatures, dust, and strong electrical fields.

Packaging. If you cannot easily store or handle the software in its original packaging, you may need to repackage it. Software can be packaged in three-ring binders with plastic sleeves, specially constructed software boxes, or paper jackets. Packaging should allow enough room to include the program as well as its documentation. Whenever possible, you should try to standardize packaging to maximize shelf space. It is also a good idea to store backup copies of software in a separate package and location.

Labeling. It will be necessary for you to put identification labels on all software, documentation, and packaging; these are three components that can easily be separated. Press-on paper labels listing the contents, subject, and call number of a program should be applied to each component. Checkout cards and checkout pockets will also need to be labeled and affixed to each software package (if you decide to use a card system for circulating materials). You may also wish to prepare copyright labels that warn against illegal copying (see Figure 6.2 for information on the software copyright law).

Software and the Law

The present copyright law for software clearly states that making a copy of a computer program for archival purposes does not violate the author's copyright for the program. The terms "archival" and "backup" are synonomous; however, copying a program for backup purposes and pirating are *not* synonomous. Pirating is the creation or distribution of illegally made copies, and software pirates are both civilly and criminally liable under copyright laws.

Permission to make legal copies is given by Section 117 of the Copyright Act; however, there are specific limitations and conditions on this permission. When can you legally make a backup copy of a program? The statute authorizes legal copying under the conditions listed below.

1. A program may be legally copied if you own the copy from which the new copy is made.

2. A copy of a program must be for archival purposes only. You may not make a backup copy of a program and use both the original and backup copy.
3. All archival copies must be destroyed in the event that you no longer rightfully own the first copy from which the archival copy was made.

The copyright law is not the only legal barrier to making backup copies. Publishers do not usually sell a program outright. Typically, software publishers sell only a license to use the program. Some software licenses restrict the purchaser from making copies of a program. There, although making a backup copy of a program may be legal according to the Copyright Act, the making of a copy may violate a vendor's licensing policy. It's a good idea to read a program's license policy before you purchase a program.

Several interesting developments in the marketplace in response to illegal copying include:

- *Mandated backup copies.* Laws in some states now require that backup copies be provided to consumers either at a low cost or as part of the original purchase price.
- *Multiple machine licensing.* Some software publishers are granting consumers a license to copy software at a discounted rate. Such licenses are usually issued either to an individual school for use in a particular building or to an entire school district.

FIGURE 6.2

Avoid applying labels in layers because the operation of the software may be affected. You should always prepare labels prior to placing them on software; however, if you do write on a label that is already affixed to software, be sure to use a felt-tip pen—using a pencil or ball-point pen will damage the disk.

Circulation. A circulation system is necessary to maintain a record of where materials can be located at all times and to determine if the software package is complete when it is returned. You will also want to know when software packages are checked out and when they are due.

If software is checked out from a lab or classroom for use in the same location, a simple written list of borrowers can be maintained to keep track of programs. You may also require the borrower to leave some form of personal identification at the circulation desk.

When software is going to be used at home or elsewhere away from the school, the software should be checked out in a protective package and should include a pocket with a circulation card.

Finally, you will need to consider who is going to staff the circulation desk. Parents and students can be trained to assist borrowers, shelve software, and monitor software circulation.

ORGANIZING THE LEARNING ENVIRONMENT

Whether computers are used in the classroom, computer lab, or library media center, educators must consider certain physical and instructional factors that contribute to their optimal use.

Modes of Computer Instruction

There are two distinct ways that software and computers can be used to support learning: (1) as an adjunct to reinforce, enrich, or illustrate material that has been previously covered via noncomputer instruction, or (2) as a replacement for regular classroom instruction. Both approaches are more easily accomplished if teachers and students have access to computers and software on a frequent basis.

Under ideal circumstances, each class participating in a computer-extended curriculum has one or more computers; however, in many schools the availablility of computers is still quite limited. Even so, careful scheduling and organization of a limited number of computers can contribute to the quality of a student's learning experience.

Three common modes of using computers are described below.

Free-Choice Activities. This approach allows the student to use the computer for whatever activity he or she chooses. Free-choice activities are frequently used as a reward for those who complete their seatwork early.

To maximize the use of each student's instructional time at the computer, a list of suggested activities or problems should be posted in the learning center. Such activites should be short enough for students to complete in a limited amount of time. Programming problems, word-processing exercises, database activities, and instructional games are examples of possible free-choice activities.

Directed Individual Activities. This mode is often used for tutorial applications and some drill-and-practice activities. To ensure the student's enjoyment and acceptance when he or she is assigned to an individual activity, the teacher should make sure that the student knows where the software is located and the exact procedure involved in using the material on the computer. The teacher must also establish an overall expectation that the activity is a reward rather than a penalty.

To make effective use of directed individual computer activities, the teacher must remember to maintain contact with the student! An occasional word of encouragement, a smile, or a direct question to the student will indicate that the teacher is aware of the student's activity at the computer. More important, the teacher must periodically provide the student with feedback about his or her work at the computer. Positive reinforcement can be given in the form of charts or checklists showing recognition of mastery.

The teacher's role in preparing students for a successful learning experience at the computer includes (a) teaching preskills necessary for understanding the concepts or tasks presented in a program and (b) providing off-computer tasks that will allow the student to practice and generalize skills learned at the computer.

Group Activities. For groups of students, one computer can be used as an "electronic chalkboard" to illustrate or enrich specific instructional objectives. For example, a teacher might use a graphing program to illustrate a group's response to a set of survey questions. Programs such as simulations allow students to work together in teams when a limited number of computers are available.

The teacher's role in preparing students to use a computer as a group for problem-solving activities involves the following stages:

- *Screener.* The teacher carefully screens the program with a small group of students to see how the students react to the program and to prepare for future problems or questions.
- *Leader.* In order to arouse the students' interest, the teacher presents the entire group with a demonstration of the program and provides information concerning the content, structure, and goals of the lesson. This is also an appropriate time for the teacher to provide the students with hints or strategies for solving problems presented in the program.
- *Manager.* The teacher preassigns students who will work together and outlines the responsiblities the group must share. In a group of four, student responsibilities might include a leader, recorder, reader, and typist. At this point, rules about how the group will work together are established.
- *Facilitator.* Once the students begin using the program, the teacher acts as a facilitator. After the group(s) have completed a program, the teacher conducts a debriefing session to discuss the results, summarize what each group has learned, and present "what if" questions to the students.

When one computer is used with a large group of students, the computer system must be placed in a location that allows the video monitor to be seen easily by the class. A large (21-inch to 26-inch) video monitor may be placed in front of the students or several smaller monitors may be linked and placed around the room.

Remember that a video monitor can distract students during a discussion, lecture, or demonstration; thus, it is helpful to have a master switch or console that permits the teacher to interrupt the video signal to gain the class's attention if necessary.

Physical Factors

Many physical factors must be considered when organizing computers for instructional purposes. For example:

1. Make sure that the lighting around computer system does not cause a glare on the monitor.
2. Carpeting in a classroom where computers are used should be treated to prevent the buildup of static electricity. Microprocessors can be damaged by voltages as low as 10 or 15 volts. If the humidity and temperature are low outside and a carpet is dry inside, it is possible for an individual walking on the carpet to build up a charge of 20,000 to 30,000 volts in his or her feet!
3. To facilitate typing at a computer, place the keyboard on a surface three to four inches lower than that normally used for writing.
4. Each computer station should have enough room for two students and their supplies.
5. If sound-producing software is used, headphones should be provided to minimize sounds that might distract others in the room.
6. Computers and software must be protected from dust and extreme temperatures. Dust covers should be provided for each computer, and software disks should be stored in their jackets. Keep computers and disks away from the chalkboard!
7. Security devices (such as locking cables) should be used to prevent possible theft or vandalism of equipment.
8. Cords and cables should be neatly arranged, and tape or special metal or plastic conduits should be used to protect them. Plug peripherals into a multiple outlet power strip to eliminate extension cords.

Instructional Factors

There is no set formula that will ensure instructional success with microcomputers. Implementing software in the classroom requires time, careful planning, and expertise in specifying instructional content to meet the needs of individual learners. Some examples of skills and information that students must learn prior to using computer software are described below.

Locate and Use Letters, Numbers, Shift Keys, and Specialized Keyboard Functions. A number of excellent software programs are available to familiarize students with the layout of a computer keyboard. One strategy is to provide students with a paper template of a computer keyboard so that the teacher can direct their attention to specific keys on the templates by pointing to keys on a transparency or large wall chart.

Develop Correct Keyboarding Skills. Keyboarding is the term used to describe the activity of entering information on the keyboard of a computer. Keyboarding instruction should teach students about correct posture at the microcomputer and the proper technique for striking the keys. Many software programs are available for teaching keyboarding skills. Old typewriters or simulated typing on keyboard templates are possible alternatives for keyboarding practice when a computer is not available.

Demonstrate Proper Use of Software and Hardware. Students must be taught to handle materials with care—not fear. One approach for training students to work with computers is to prepare a series of instructional modules that cover such topics as how to locate, check out, and load a program. The modules can be self-taught or taught by a teacher or peer tutor. Schools that use this approach often require students to pass a computer proficiency test. When a student passes the test, he or she receives a license or authorization card that must be presented to a teacher or aide prior to using a computer.

Computer Terminology. Instruction involves communication, and instruction about computers involves a special vocabulary. Students must be taught the meaning of common computer terms. For example, a student must recognize what the teacher means when he or she says, "Boot-up a disk, then move the cursor next to the item you wish to select from the menu."

Management Rules. It is a good idea to establish schoolwide guidelines for computer usage. A set of guidelines for computer use posted in the area where students will be working with computers is also recommended. Guidelines should address such issues as:

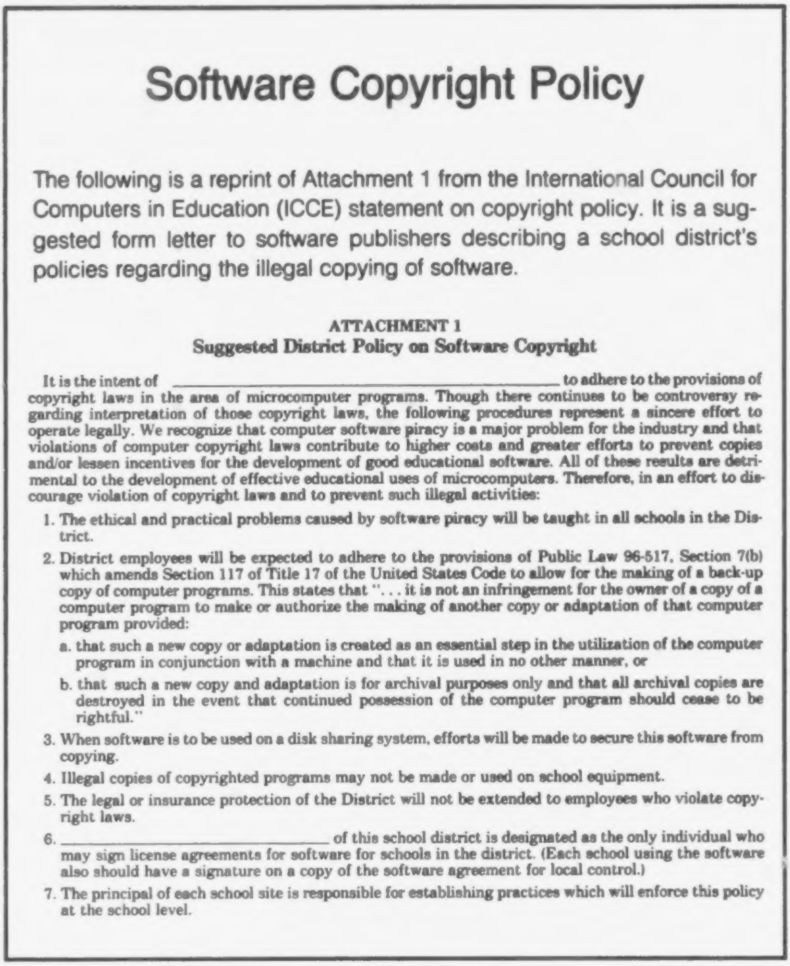
1. A list of software that students can check out.
2. How to load and operate a computer or printer.
3. The length of time allowed for one person to work with a computer.
4. The number of students who may work together at a computer.

Copyright Laws. Some experts believe that software piracy has reached epidemic proportions because of ignorance of the law. Educators have an obligation to expose students to the legal, ethical, and practical problems caused by the illegal copying of software. Making illegal copies of copyprotected and copyrighted software is stealing. It doesn't matter if the copying is performed for profit or as a favor for a friend—it is still stealing. This applies to illegal copying by educators as well as students.

Students should know the software copyright policy of the district or school. The rules regarding copyrighted software are generally the same as those pertaining to printed materials. An excellent model for a district policy on software copyright is published by the International Council for Computers in Education (ICCE). Figure 6.3 shows Attachment 1 from the ICCE statement on software copyright policy.

The Surrounding Environment

A computer center should be more than just a room with one or more computers. Bright, cheerful, and well-organized surroundings are factors that contribute to any successful learning envi-



Software Copyright Policy

The following is a reprint of Attachment 1 from the International Council for Computers in Education (ICCE) statement on copyright policy. It is a suggested form letter to software publishers describing a school district's policies regarding the illegal copying of software.

ATTACHMENT 1
Suggested District Policy on Software Copyright

It is the intent of ______________________________ to adhere to the provisions of copyright laws in the area of microcomputer programs. Though there continues to be controversy regarding interpretation of those copyright laws, the following procedures represent a sincere effort to operate legally. We recognize that computer software piracy is a major problem for the industry and that violations of computer copyright laws contribute to higher costs and greater efforts to prevent copies and/or lessen incentives for the development of good educational software. All of these results are detrimental to the development of effective educational uses of microcomputers. Therefore, in an effort to discourage violation of copyright laws and to prevent such illegal activities:

1. The ethical and practical problems caused by software piracy will be taught in all schools in the District.
2. District employees will be expected to adhere to the provisions of Public Law 96-517, Section 7(b) which amends Section 117 of Title 17 of the United States Code to allow for the making of a back-up copy of computer programs. This states that ". . . it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
    a. that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
    b. that such a new copy and adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful."
3. When software is to be used on a disk sharing system, efforts will be made to secure this software from copying.
4. Illegal copies of copyrighted programs may not be made or used on school equipment.
5. The legal or insurance protection of the District will not be extended to employees who violate copyright laws.
6. ____________________ of this school district is designated as the only individual who may sign license agreements for software for schools in the district. (Each school using the software also should have a signature on a copy of the software agreement for local control.)
7. The principal of each school site is responsible for establishing practices which will enforce this policy at the school level.

FIGURE 6.3

ronment. A computer work area should be equipped with other important teaching tools such as an overhead projector, screen, and chalkboard.

Bulletin boards can be used to post schedules, instructions, special projects, awards, and examples of student work related to computer use.

Specific instructional topics may require the availablility of other important teaching supplies. Reference books, pencils, paper, calculators, kits, games, scientific instruments, and other commonly used instructional tools should be readily available in a computer learning center.

Finally, if there is a need to establish a quiet work area or privacy for students who are working at a computer, bookcases, file cabinets, and study carrels can be used as area dividers.

Equipment Placement

There are several considerations regarding the placement of computer equipment. If a school has a lot of computers, several computer labs might be created and a computer placed in each classroom. In such an ideal setting, a computer lab should be available for use by teachers in all subject areas, and scheduling of the lab should be flexible.

An important point to remember is that learning with computers is not always so much a question of equipment placement as one of the appropriate use of the equipment that is available. Therefore, an important task for any school staff is to assess how their computers will actually be used. The distribution of computers in a school should change with time, so it is also important to reevaluate placement decisions periodically.

The major issue in regard to computer placement usually is whether to centralize all of the school's computer resources in a lab or to distribute the available resources throughout the school. Centralizing computer equipment in a school requires the presence of a supervisor. A lab supervisor can help answer questions, schedule computer use, catalog software, and maintain equipment. Placing a single person in charge of all computers in a school can become a problem. The lab supervisor who projects an aura of having "all the answers" may discourage other teachers from tackling their own computer projects

From a funding perspective, a computer lab has some advantages. For example, a lab setting permits computers to be networked—a number of microcomputers are connected to a central disk drive, which can run a single program (or, with some systems, two or more programs) for each computer at the same time. In some instances networking can reduce the costs of software and disk drives; however, many educational software programs cannot be used when computers are networked.

Instructionally speaking, a computer lab can make teaching easier because the students can all be taught the same skills at the same time. Another advantage of a computer lab is that students can use the computer for a sustained period of time. Frequent and sustained work with computers is more likely to result in learning gains than sporadic use.

There are times, however, when a teacher will want to use a computer on the spur-of-the-moment. The teacher's flexibility to demonstrate a concept, conduct small-group activities, or assign immediate remediation and practice via the computer is lost when computers are not readily available in a classroom.

Scheduling and Grouping

When students work together on a computer, the teacher should preassign the students to groups, being careful to avoid creating a group that promotes the confidence of only one student. Students who work together on a computer should be encouraged to work as a team. It is also important for educators to convey that all students have the right to use computer technology. Teachers should be sensitive to the possiblility of creating stereotypes that say computers are reserved for good students, boys, or nonhandicapped individuals.

When a limited number of computers are available, free-choice activities should be scheduled so that student interruptions are minimized for the teacher. When the software has been loaded into the computer, the teacher can schedule each student at the computer for a specified period of time, using a kitchen timer to keep track. Each scheduled student can reset the timer and proceed with his or her activity.

A block-time format works well for scheduling students to complete directed individual activities. Each student in the class is assigned a period of time to use the computer by posting a chart that specifies pupil names and their computer access times. In a typical twenty-five-hour week of classroom instruction, a class of twenty-five students can easily be allotted twenty to thirty minutes each per week at the computer.

In a lab setting, all teachers in a school might plan and coordinate their use of the facilities. For example, English classes might use the lab the first month, math classes the second, and computer literacy classes the third. An alternate approach is to schedule a lab by periods, with English the first period, math the second, and so on. These are not ideal scheduling solutions, but they do permit the maximum use of computer equipment by various classes throughout a school.

CONCLUSION

The explosive growth of microcomputers in education has raised a number of issues without precedent. There is no legacy of computer use in the schools that can be referred to as a standard upon which to base decisions. What then are the salient features to remember when organizing and using educational software? First, regardless of where the computer is located, attention must be given to keeping the equipment in use. Second, a number of instructional and physical factors must be considered when using computers with students. Finally, the software selection process—the focus of this book—must not be taken lightly. Educators must think clearly about their instructional objectives, what computers can do to help, how that assistance can be accomplished, and if there are more effective and less expensive alternatives to using a microcomputer program.

READINGS AND RESOURCES

"ICCE Policy Statement on Network and Multiple Machine Software." *The Computing Teacher* 11 (1983):18–22.

This is a comprehensive statement calling for more cooperation between educators and software developers relative to the problem of software piracy.

REFERENCES

Goodson, B., and Lathrop, A. *Courseware in the Classroom*. Menlo Park, CA: Addison-Wesley Publishing Company, 1983.

PROJECTS AND QUESTIONS

1. Ask three library media specialists or three computer teachers who use computers on a regular basis to answer the questions listed below. Summarize their answers in a two-page report.
    a. What system is used to catalog, process, and check out software in your school? What are the strengths and weaknesses of the system?
    b. What policies have been established in your school regarding student, teacher, and administrative access to computers? What procedure was used to establish the policies?
    c. Where are the computers located in your school? What rationale and process was used to select the location(s)?
2. Prepare catalog cards for three different microcomputer programs.
3. Create a wall chart for use in a library media center or classroom for one of the following topics:

   a. The steps for loading a computer.
   b. A set of rules for computer use and student behavior.
   c. A list of useful computer terms and their definitions.
   d. An explanation of how to read and understand a catalog card for computer software.
   e. A picture of a computer keyboard for use in teaching students where specific keys are located on a computer.

4. Draw and label a picture that shows how you would organize an ideal computer lab, classroom, or library media center.
5. Describe five physical and five instructional factors that must be considered in order to use educational software effectively.
6. Predict five ways that computers may affect how schools are physically and instructionally organized in the future. (Look at your "crystal ball" and let your imagination take over).
7. Describe a situation in which you could use computer software, including a description of the audience, the objectives, the role of the computer, and the expected advantages of using the computer.
8. Use an educational software program in your teaching. Describe the software, its objectives, results, and reactions (both yours and your students') to the software.

Index

Abacus, 19
Aiken, Howard, 20
Alpha-testing, 4
American Association of School Librarians, 162
Apple, 14, 74
Atkinson, Richard, 26

Babbage, Charles, 20
Backup policy, 7, 10, 133, 166–167
BASIC programming language, 3, 84–86
Beta-testing, 4
Blinking and flashing of text, 143–144
Branching, 33, 36
Brigham Young University, 25
Bruner, Jerome, 43

Cassette, 22
Cathode-ray tube, 23
Center for Learning Technologies (New York), 14
Center for Social Organization of Schools (Johns Hopkins University), 3
Color, use of, 23–24
Command summary card, 135, 136
Commodore, 14
Computer assisted instruction (CAI), 25 (*see also* Instructional games; Simulation-oriented CAI; Tutorial CAI)
  limitations of, 96
  modes of, 168–170
Computer games, 49–55
Computer technology, changes in, 1
Computers:
  classroom management of, 168–177
  classroom time spent with, 3
  demonstrating careful use of, 172
  history of, 19–20
  home vs. school, 13
  instructional factors in using, 171–173
  memory capacity of, 22
  physical factors in using, 171
  placement of, 175–176
  selection of, 25
  sources of training in, 12
  and teacher education, 11–13
Computers in Teaching Mathematics, 55
The Computing Teacher, 97, 98–99
Consumers Union, 97, 104–105
Copy frame, 34
Copy protection schemes, 7, 173
Copyright Act, Section 117, 166, 174
Copyright laws, 166–167, 173
CRT, 23

Daisy wheel, 24
Dartmouth College, 85
Database programs, 69–71
  features of, 70–71
Dedicated word processor, 63
Dewey Decimal Classification System, 163
Disadvantaged students, 26
Disk drive, 23
Diskette, 7, 8–9, 22
  life of, 162
Documentation, 134–137
Dot matrix printer, 24
Drill and practice, 26–32
  criticisms of, 30
  designs of, 27–28
  effectiveness of, 32
  history of, 26
  value of, 31–32

Eckert, J. Presper, 20
Educational Products and Information Exchange (EPIE), 4–6, 10, 26, 97, 104–105, 122
The Educational Software Selector (*see* TESS)
Egrule frame, 35
80-column programs, 64–65
Electronic Learning, 97, 100–101
"Empire," 42
ENIAC, 20
EPIE (*see* Educational Products and Information Exchange)
Error correction, 144–145

Field testing, 4
Floppy disk, 22, 63 (*see also* Diskette)
Frames, 33, 34–35

Gradebook programs, 81–82
Graphing programs, 78–79

Huntington Projects, 43

IBM, 14, 63
ICAI, 37
Icons, 74
Information processing tools, 69–72
Instructional games, 49–55
  criticisms of, 54
  designs of, 50
  effectiveness of, 54
  examples of, 52–54
  history of, 50
  value of, 54
  violence in, 54
Instructional objectives, 137
Integrated programs, 74
Intelligent CAI (ICAI), 37

International Council for Computers in Education, 173, 174

Johns Hopkins University, 3

K, 21–22
Kemeny, John, 84
Keyboard familiarity and skills, 172
Kilobyte, 21–22
Koala Pad, 73–74
Kurtz, Thomas, 84

Language arts, 26
Learner Verification and Revision (LVR), 4–6
Letter-quality printer, 24
"Life," 42
LOGO programming language, 87–88
LVR, 4–6

MacIntosh, 74
Magnetic Tape Selectric Typewriter, 63
Mainframe computers, 21
Malone, Thomas, 55
MARK I, 20
Mathematics, 26
Mauchly, John, 20
Memory (*see* RAM)
Menus, 137, 140
Microchips, 20
Microcomputers, 21, 22 (*see also* Computers)
MicroSIFT, 97, 102, 122, 124–129
MicroSIFT Evaluator's Guide for Microcomputer-Based Instructional Packages, 122, 124–129
Minicomputers, 21 (*see also* Computers)
Modem, 103
Monitors, 23–24
  for word-processing, 65
"Monopoly," 42

National Commission on Excellence in Education, 15
Networking, 175–176

Papert, Seymour, 87
Pascal, Blaise, 19, 20, 86
Pascal programming language, 86
Peripherals, 21
Piaget, Jean, 50
PILOT programming language, 87, 88
"Planet Management," 42
PLATO (Programmed Learning for Automatic Teaching Operations), 25
Presentation rate, 141
Printers, 24
Program, defined, 84
Programmed instruction, 33
Programming languages, 84–89
  reasons for teaching, 89
  selecting, 89
Prompts, 72, 143
Public domain software, 107
Puzzle- and poster-making software, 82–83, 84, 85

Radio Shack, 14
RAM, 21–22
Random access memory, 21–22
Range of student responses, 146
Readability analysis programs, 79–81
Record-keeping of student scores, 147–150
RICE (Resources in Computer Education), 106
Ruleg frame, 35

Scheduling and grouping of students, 176–177
Screen design, 137, 141
Sears List of Subject Headings, 163, 164
Simulation-oriented CAI, 42–49
  designs of, 43–44
  effectiveness of, 49
  history of, 42–43
  limitations of, 48
  value of, 48–49
Skinner, B. F., 33
Software:
  backup policies, 7, 10, 133, 166–167
  careful handling of, 172
  on cassettes, 22
  cataloging of, 163–165
  categories of, 61
  circulation of, 167–168
  compared to other curriculum materials, 2
  for the creative arts, 72–74
  database programs, 69–71
  defined, 1
  directory of publications reviewing, 113–118
  directory of publishers offering previews, 109–113
  on diskettes, 22
  drill and practice, 26–32
  evaluation and review process, 122–123, 130–132
  evaluation forms for, 122, 124–129
  evaluation questions for, 132–153
  field testing of, 4–6
  flexibility of, 2–3
  gradebook programs, 81–82
  graphing programs, 78–79
  integrated, 74
  instructional content of, 146–147
  labeling for library use, 165
  machine compatibility and, 13–14, 96

  market by subject, type, and grade level, 26–27
  obtaining review copies of, 106–107
  on-line computer searches for, 103
  organizing, 161–168
  packaging for storage, 165
  piracy of, 7, 166–167, 173
  producers of, 14
  protection of, 7, 173
  public domain, 107
  puzzle- and poster-making, 82–83, 84, 85
  readability analysis programs, 79–81
  reviews of, 96–97, 98–102
  selection criteria for, 121–122
  selection process, 10, 95–108
  sources of, 6–7
  spelling checkers, 77–78
  spreadsheet programs, 71–72
  storage of, 165
  as tools, 61
  warranties for, 133, 135
  word-processing, 62–68
Software by title. (*Page numbers in italics give machine compatibility, manufacturers' addresses.*)
  Add-Your-Own-Content, 151, 153, *156*
  Alien Addition, 28–29, *57*
  Alphabet Keyboard, *102*
  Apple Logo, *91*
  Apple Writer IIe, 66, 67, 68, *91*
  Appleworks, 74, *91*
  Bank Street Writer, 66, 67, 73, *90*, 143, *156*
  Capitalization, 150, *156*
  Crossword Magic, 82, 83, *91*
  Custom Ware, 141, 149, *156*
  Delta Drawing, 135
  Discovery Lab, 44, 46, *58*
  Friendly Filer, 69–70, *92*
  Gertrude's Puzzles, *98*–99
  Kidwriter, *100*–101
  Know Your Apple IIe, 38, *58*
  Koala Micro Illustrator, 73–74, *91*
  MacWrite, 74, 75, *91*
  Mastertype, 52, *58*
  Microtype, 38, 39, *58*
  Microzine, 133, 135
  Moptown Hotel, 53
  Music Theory, 29–30, *57*
  Nouns and Pronouns, *156*
  Number Farm, 53–54, *57*
  Oh, Deer!, 44–45, 151, 152, *156*
  Oregon, 46–47, *58*
  Pac Man, 50
  PFS:Graph, 79, *91*
  The Printshop, 83, 85, *91*
  Readability Tests, 80, *91*
  The Report Card, 81–82, *91*
  Rocky's Boots, 46–47, *58*
  Sensible Speller, 77, *91*
  Songwriter, 72–73, *91*
  Space Invaders, 50
  Spell It!, 153, 154, *156*
  Storytree, 67, 69, *90*, 137
  Superpilot, 87, 88, *91*
  Trap-a-Zoid, 142, *156*
  Using a Calendar, 148, 153, 154, *156*
  Word Find, 82, 84, *91*
  Writing a Character Sketch, 38, 39, *58*
  Writing a Narrative, 133, 134, *156*
Software companies (*see also under* Software by title)
  Developmental Learning Materials (DLM), 28
  Minnesota Educational Computing Corporation (MECC), 29, 38, 44, 46
  Scholastic Software, 133
Sound, 151
Spelling checkers, 77–78
Spreadsheet programs, 71–72
Stanford University, 26
Student interest, 150–151
Suppes, Patrick, 26

Teaching machines, 33
TESS, *The Educational Software Selector*, 26, 103
TICCIT (Time-shared Interactive Computer-Controlled Instructional Television), 25
Tool software
  general-purpose, 62–74
  kinds of, 62
  special-purpose, 74–83
  tool-making, 83–89
Turtle graphics, 87–88
Tutorial CAI, 32–42
  designs of, 33
  effectiveness of, 41–42
  examples of, 38–39
  history of, 33
  limitations of, 40
Tutorials in documentation, 136–137, 138–139
Typing, amount required in programs, 142

University of California at San Francisco, 87
University of Illinois, 25

Video games, 50 (*see also* Instructional games)

Wirth, Niklaus, 86
Word processing, 62–68
  creative writing and, 67
  features of, 63–66
  history of, 62–63
Word wrap, 65

ALSO OF INTEREST

MICROCOMPUTERS IN THE ELEMENTARY CLASSROOM

A Guide for Teachers
George W. Bright

All about using computers to improve elementary classroom instruction…and how to integrate them into existing programs.

SOLVING DISCIPLINE PROBLEMS

Strategies for Classroom Teachers, Second Edition
Charles H. Wolfgang and Carl D. Glickman

Today's eight leading approaches to classroom discipline, described and demonstrated for practical use.

HANDBOOK FOR EFFECTIVE DEPARTMENT LEADERSHIP

Concepts and Practices in Today's Secondary Schools, Second Edition
Thomas J. Sergiovanni

Shows how to use over 100 practical aids plus today's expert management thinking to work faster and smarter for better schools.

EVALUATING INSTRUCTIONAL PROGRAMS

Second Edition
Bruce W. Tuckman

A comprehensive guide to evaluating programs without necessarily hiring experts — including a tested four-step procedure, detailed case studies, and model forms and instruments.

For more information on these and other titles, write to:

ALLYN AND BACON, INC.
Longwood Division
7 Wells Avenue
Newton, MA 02159

ISBN 0-205-10468-1
EDP 2304686